University of Louisville
ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

5-2017

Vehicle make and model recognition for intelligent transportation monitoring and surveillance.

Faezeh Tafazzoli
University of Louisville

Follow this and additional works at: https://ir.library.louisville.edu/etd

 Part of the Computer Engineering Commons

Recommended Citation

Tafazzoli, Faezeh, "Vehicle make and model recognition for intelligent transportation monitoring and surveillance." (2017). *Electronic Theses and Dissertations.* Paper 2630.
https://doi.org/10.18297/etd/2630

This Doctoral Dissertation is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

VEHICLE MAKE AND MODEL RECOGNITION FOR INTELLIGENT TRANSPORTATION MONITORING AND SURVEILLANCE

By

Faezeh Tafazzoli
M.Sc., Computer Science, University of Nevada, Reno, 2012
M.Sc., Computer Engineering, Amirkabir University of Technology, Iran, 2008

A Dissertation
Submitted to the Faculty of the
J.B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy in Computer Science and Engineering

Department of Computer Engineering and Computer Science
University of Louisville
Louisville, Kentucky

May 2017

Copyright 2017 by Faezeh Tafazzoli

All rights reserved

VEHICLE MAKE AND MODEL RECOGNITION
FOR INTELLIGENT TRANSPORTATION MONITORING AND
SURVEILLANCE

By

Faezeh Tafazzoli
M.Sc., Computer Science, University of Nevada, Reno, 2012
M.Sc., Computer Engineering, Amirkabir University of Technology, Iran, 2008

A Dissertation Approved On

April 24, 2017

by the following Dissertation Committee:

Hichem Frigui, Ph.D., Dissertation Director

Amir Amini, Ph.D.

Olfa Nasraoui, Ph.D.

Juw Won Park, Ph.D.

Hui Zhang, Ph.D.

TO ALL MY LOVED ONES

Nasser for his love, my parents for their sacrifices, and my sister for her supports

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the support of many people. I would like to express my deep gratitude to my advisor Dr. Hichem Frigui for giving me the precious opportunity to pursue my doctoral studies under his guidance. He provided me with the perfect balance of guidance and freedom. His broad views on science and his intellectual creativity and analysis have been of great inspiration for my research work.

Thanks to my reading and oral committee members, who have provided invaluable feedback in the preparation of this document and otherwise throughout the years: Dr. Olfa Nasraoui, Dr. Amir Amini, Dr. Juw Won Park and Dr. Hui Zhang. Their critical questions, especially about my thesis proposal, have guided me in the right direction and are directly reflected in this dissertation.

Thank you to my friends and colleagues in the Multimedia Research Lab, and the Computer Engineering and Computer Science Department, who have been the source of countless support and encouragement, and more generally, made the day to day of my PhD downright enjoyable.

During my PhD, I had the chance to spend a few internships at Xerox Research Center, PARC, and I am profoundly grateful to Bob Loce for inviting me there and believing in me. Many thanks to all other members of XRCW for a warm welcome and providing me with a rewarding experience.

I am eternally grateful to my family: my parents for their unconditional support and encouragement to pursue my interests, even when the interests went beyond boundaries of language and geography; my wonderful sister for taking care of everything while I was away and for understanding me. It's the passion for exploration and adventure combined

with determination and hard work that I learned from you. Those values are what led me through my PhD and let me handle the every single moment of missing you all.

Endless gratitude is for my beloved Nasser, for being the travel partner of my life and for his love, understanding, and continuous support. Thanks for believing in me, inspiring me, and cheering me up at the hard times. The PhD journey has been incredibly beautiful sharing it with you.

ABSTRACT

VEHICLE MAKE AND MODEL RECOGNITION
FOR INTELLIGENT TRANSPORTATION MONITORING AND SURVEILLANCE

Faezeh Tafazzoli

April 24, 2017

Vehicle Make and Model Recognition (VMMR) has evolved into a significant subject of study due to its importance in numerous Intelligent Transportation Systems (ITS), such as autonomous navigation, traffic analysis, traffic surveillance and security systems. A highly accurate and real-time VMMR system significantly reduces the overhead cost of resources otherwise required. The VMMR problem is a multi-class classification task with a peculiar set of issues and challenges like multiplicity, inter- and intra-make ambiguity among various vehicles makes and models, which need to be solved in an efficient and reliable manner to achieve a highly robust VMMR system.

In this dissertation, facing the growing importance of make and model recognition of vehicles, we present a VMMR system that provides very high accuracy rates and is robust to several challenges. We demonstrate that the VMMR problem can be addressed by locating discriminative parts where the most significant appearance variations occur in each category, and learning expressive appearance descriptors. Given these insights, we consider two data driven frameworks: a Multiple-Instance Learning-based (MIL) system using hand-crafted features and an extended application of deep neural networks using MIL. Our approach requires only image level class labels, and the discriminative parts of each target class are selected in a fully unsupervised manner without any use of part annotations or segmentation

masks, which may be costly to obtain. This advantage makes our system more intelligent, scalable, and applicable to other fine-grained recognition tasks.

We constructed a dataset with $291,752$ images representing $9,170$ different vehicles to validate and evaluate our approach. Experimental results demonstrate that the localization of parts and distinguishing their discriminative powers for categorization improve the performance of fine-grained categorization. Extensive experiments conducted using our approaches yield superior results for images that were occluded, under low illumination, partial camera views, or even non-frontal views, available in our real-world VMMR dataset. The approaches presented herewith provide a highly accurate VMMR system for realtime applications in realistic environments.

We also validate our system with a significant application of VMMR to ITS that involves automated vehicular surveillance. We show that our application can provide law inforcement agencies with efficient tools to search for a specific vehicle type, make, or model, and to track the path of a given vehicle using the position of multiple cameras.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Nowadays, dealing with large amounts of data is a very active research topic. The world is continuously generating countless data. Processing and extracting information from these data has become a challenging task. Images and videos make up a large portion of this data. This is due in part to the fact that there are over one billion smart phone users around the world who take photos and videos every day. Consequently, on Facebook alone, more than 250 billion photos have been uploaded and, on average, it receives over 350 million new photos every day [1]. Similarly, YouTube reports that 400 hours of video are uploaded every single minute [2]. Additionally, there is the data that have not made it into the Internet (yet), such as the 24/7 video feeds from millions of surveillance cameras in parking lots, convenience stores, ATMs, airports, etc. How to effectively understand and use this data is a critical task.

As one of the fundamental topics in computer vision, *image classification* serves as a basic way to organize images in an unsupervised way. It involves determining whether or not an image contains some specific class of objects. The high intra-class variability and the changes in global appearance of the objects within the same category have been the main challenges in basic recognition tasks. Ideally, a category model should be able to represent all the objects within the category, and be flexible enough to accommodate their intra-class variability.

Recently, driven by real-life applications, subordinate-level categories, such as identification of different species of birds [3–5], flowers [6, 7], leaves [8], breeds of dogs [9] and models of aircraft [10], has become an important direction of research in image classification. These

problems which directly address the intra-class variability are regarded as a *fine-grained visual classification* task which attempt to classify same-category objects with only subtle distinctions [11]. In essence, the main difference between basic-level and fine-grained visual classification is the level of difference between different categories. Distinguishing a bird from an airplane is a relatively easy task as there are plenty of helpful visual cues. However, this is not the case when we want to differentiate between two species of birds with a minute appearance difference such as different colors of beak. The latter problem requires methods that are more discriminative than those used for general image classification.

Two major challenges need to be tackled in fine-grained classification. First, differentiating subtle details in appearance which would define the target fine-grained classes. This calls for an appearance feature representation that retains details critical for discrimination and discards unnecessary information. Another challenge is in discovering and locating the parts that contain discriminative details. In such applications, a reliable classifier would amplify the differences while suppressing the common features in the representation of different categories. Despite these challenges, however, fine-grained tasks can be very useful when properly addressed. It can be considered as one of the cornerstones in computer vision due to its potential to make computers rival human experts in visual understanding in many real-world applications.

A very interesting and challenging category of images with considerable intra-class variability and inter-class similarity are vehicles. Cars are now indispensable from our modern life which has made their classification an appealing problem because it is a well-defined fine-grained recognition task that plays an important role in many potential applications.

1.1.1 Intelligent Transportation Systems

Over the recent years, a deluge of innovative technologies and solutions are bringing Intelligent Transportation Systems (ITS) closer to reality. The development of digital image sensors and computer vision techniques offer a great deal of advantages in enabling many important ITS applications and components such as Advanced Driver Assistance Systems

(ADAS), Automated Vehicular Surveillance (AVS), traffic and activity monitoring, traffic behaviour analysis, traffic management, intelligent parking, and self-guided vehicle systems. Other deployments are found both in public and commercial sphere. Identification and classification of vehicles is of great interest in these applications, due to elevated security concerns in ITS and demanding areas such as targeted advertisement or surveillance for crime prevention and safety.

In 2014, there were 907 million passenger vehicles and 329 million commercial vehicles registered worldwide, in comparison with 2006 statistics which had 678 and 248 million passenger vehicles and commercial vehicles, respectively [12]. This shows an increase of 33.7% in passenger vehicle numbers and 32.6% in commercial vehicles. If this trend increases with the same pace, in 2035 there will be approximately 1.7 billion registered vehicles on road worldwide. This brings up a need to implement a system which can identify the vehicles effectively and accurately and employed in different scenarios of incident detection, behavior analysis and understanding. Generally, the problems of vehicle detection, identification, classification, and tracking can all be defined in form of AVS. By applying fine-grained classification of vehicles in transportation and public security, we can acquire more meta information like vehicle make, model, logo, production year, max speed, acceleration, etc. Over the years, significant research has been done to solve challenges in these areas [13,14]. However, classifying vehicles into fine categories such as makes and models, has gained attention only recently, and many challenges remain yet to be addressed [15–17].

The focus of this dissertation is on developing novel approaches to address the challenges in automated Vehicle Make and Model Recognition (VMMR), utilizing state of the art computer vision-based techniques.

1.1.2 Automated Vehicular Surveillance

The need for vehicle identification and classification has become prevalent in recent years as a result of the increase in security awareness for access control systems in parking lots, buildings, and restricted areas. In such highly vulnerable areas, an AVS system run-

ning over surveillance video footage can greatly assist the security personnel in verifying vehicle from appearance given certain types, makes, models, or colors, and track it across a multiple camera network. Moreover, in post-event investigations of vehicle-related crimes, law enforcement agencies usually require searching and monitoring a suspicious vehicle from millions of traffic images based on general descriptions of car features from a victim. In such scenario, taking advantage of VMMR technology would save considerable amount of time, resources and manpower, thereby speeding up the event of crime capture. In addition, for applications such as electronic toll collection, vision-based AVS systems could serve as a complementary tool in improving efficiency of existing systems to apply different rates to different types of vehicles inexpensively and automatically. In traffic control or traffic monitoring, statistics of vehicle flow, associated with vehicle models, is more helpful in an intelligent transportation system. Travel time between monitored points can be estimated to provide the detailed traffic status during peak traffic hours.

Traditional vehicle identification systems recognize makes and models of vehicles relying on manual human observations or automated license plate recognition (ALPR) systems [18–20]. Both approaches are failure-prone and have several limitations. It is practically difficult for human observers to remember and efficiently distinguish between the wide variety of vehicle makes and models. On the other hand, the AVS systems that rely on license plates suffer from several limitations. First, most surveillance cameras are not installed for license plate capturing, thus, plate recognition performance drops dramatically on images/video data captured by these cameras. Furthermore, license plates are easy to be forged, damaged, modified, occluded, or invisible due to uneven lighting conditions. Moreover, in some areas, it may not be required to have the license plate at the front or rear of vehicle. Thus, if the ALPR system is not equipped to check for license plates at both views of the vehicle, it could fail. This could lead to retrieving the wrong information regrading make or model of the vehicle from the registry [21].

To overcome the above shortcomings in traditional vehicle identification and classification systems, the make and model of the vehicle recognized by the VMMR system can comple-

ment the license plate recognition systems by providing a higher level of robustness against fraudulent use of license plates or poor image quality and consequently further enhance security.

1.2 Vehicle Make and Model Recognition

The problem of vision-based automated vehicle classification into makes and models is an important task for AVS and other ITS applications which can be considered as a challenging multi-class fine-grained image classification problem, in which a "class" is a particular vehicle manufacture, model and year. Most works first adopt a vehicle detection step which produces Regions of Interest (ROIs) containing the vehicle's front or rear viewpoint, segmented from the background. The Vehicle classification systems then work on the features extracted from these ROIs.

1.2.1 Challenges and Issues

Vehicles offer several unique properties compared to other objects. They provide a more diverse and challenging set of issues and facilitate a range of novel research topics in fine-grained image classification. There are two broad categories of challenges in VMMR: (1) Multiplicity, and (2) Ambiguity [22].

The multiplicity problem stems from one vehicle model of the same make having different shapes and/or appearances in different years. Most of the time, a vehicle's shape is remodeled at different times to fit market requirements. Figure 1.1 displays the multiplicity problem in sample images from the VMMRdb dataset introduced in section 5.1. For instance, the differences between images of Honda Civic in years between 1986 to 2015 are in changes in the shape of tail lights, bumper or an optional addition of rear spoiler in some cases.

The ambiguity problem can be further classified into two types: (a) Inter-class similarity, and (b) Intra-class variability. The former ambiguity refers to the issue of vehicles of different manufactures having visually similar shape or appearance, i.e., two different

make-model classes have similar front or rear views. For example, "Ford Escape 2010" and "Mazda Tribute 2008" have visually similar back-view appearance (Figure 1.2). The latter kind of ambiguity is a result of similarity between different models of the same make. For example, the "Altis" and "Corolla" models of "Toyota" are visually very similar (Figure 1.2).

Another major challenge can arise when features are computed from the background. This can affect the learning mechanism. Although many other fine-grained applications might face similar issues, the considerably large number of car models, including different car manufactures and models depending on the year has made VMMR one of the most challenging fine-grained classification problems. This application, thus, can potentially foster more sophisticated and robust computer vision models and algorithms.

Figure 1.1: Multiplicity Problem with sample classes in VMMRdb dataset (section 5.1)

Figure 1.2: Ambiguity Problem with sample classes in VMMR dataset (section 5.1)

For humans, recognizing the makes and models of the cars might be a straightforward task, especially for car enthusiasts. In fact, cars can usually be identified by the human eye due to certain key aspects, such as logos, hood ornaments, or lettering. However, since most vehicles have similar global shapes, un-textured regions and their appearances are often dominated by environmental reflections and highlight lines, this has traditionally been a hard task for computers. In some cases, differentiating visually between some models of a specific manufacturer, from certain viewpoints, is quite difficult even for humans. Cars, generally, yield large appearance differences in their unconstrained poses, which demands viewpoint-aware analyses and algorithms. Thus, for an algorithm to be reliable and robust, it should only rely on fine differences in local appearance, and retain discriminative details that are highly domain-specific. Additionally, a robust object category recognition system needs to tackle varying imagery conditions that exists in different applications. Traditional methods for image classification need effective descriptors and machine learning algorithms to obtain good accuracy levels. The descriptors are used to represent an object with specific features and a classifier learns the image label based on these features. These feature representations require a significant amount of domain knowledge and typically, they do not generalize well to new domains. Additionally, these methods work under the assumption that all of the extracted features are useful for classification.

For fine-grained recognition tasks, specifically, the challenge is in discovering and locating the regions that contain these discriminative details. It has been proven in cognitive research studies [11] that basic-level recognition is based on comparing the shape of the objects and their parts, whereas subordinate-level recognition is based on comparing appearance details of certain object parts [3]. If irrelevant parts are used, it is virtually impossible to distinguish between two models of a vehicle. However, if we know the location of the discriminative regions, the recognition task becomes easier. Thereby, one important common feature of many existing fine-grained methods is that they rely heavily on annotations of an object or even object parts to learn a model that can depict the object as precisely as possible and build the correspondence between object parts [23]. This approach is clearly costly and

may be difficult to scale up to handle many different types of fine-grained classes. This can impose limitations to the application of these methods to real-world problems.

In this dissertation, we hypothesize that an efficient, yet robust solution to such fine-grained recognition tasks should entail both localizing important regions within the image with minimal supervision, and effectively describing each salient region by a feature vector. For this problem we consider two data driven frameworks: a Multiple Instance Learning-based system and a deep neural network.

1.3 Overview of Proposed Solution

In traditional supervised learning, the training set is defined by $D = \{< x_i, y_i >\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^d$ is an instance vector that characterizes object o_i. A label $y_i \in \mathbb{Y} = \{1 \dots C\}$ is associated with each x_i, where C is the number of classes. The goal is to learn a classifier $g : \mathbb{R}^d \rightarrow \mathbb{Y}$, that can predict y_t, the label of an unseen test sample x_t.

For this type of learning, domain knowledge is not required, but all the labels should be carefully provided to achieve a good performance, acceptable robustness, and generalization capabilities. However, for most applications, either the data is labeled ambiguously and at a coarse level, or a tedious manual process is needed to label the data. In the latter case, only a small set of labeled samples may be available. For example, in image annotation, large amounts of data are available and could be used for learning. Typically, only tags are provided and could be used as indicators of the existence of an object of interest within the images (e.g. vehicle, airplane, or bird). Unfortunately, despite the scalability of many recent machine learning algorithms, they still require the full engaged cognition of a human being to assign labels at a finer level, e.g. label regions within images. However, the exact location and boundary of those objects within the image is not available and is too tedious to extract for large collection of images. Thus, in this application, it is not easy to overcome the ambiguity of labeled data even by including humans in the loop to either label or identify regions of interest [24]. This is because even though with crowd-sourcing, labels can be assigned to data points quickly, easily, and cheap [25], this strategy yields

noisier data than traditional annotation since high-quality manual outlining is intrinsically ambiguous, subjective, and prone to errors (e.g. difficulty to select discriminative parts of an object within an image).

Unsupervised learning methods [26,27], on the other hand, ease the burden of manual annotation, but often at the cost of decreased performance.

In the middle of the spectrum is the weakly supervised learning scenario. The idea is to use coarse-grained annotations to aid automatic exploration of fine-grained information. One particular form of weakly supervised learning is Multiple Instance Learning (MIL). MIL is a relatively new framework that can learn with partially labeled data [28].

1.3.1 Multiple Instance Learning

Multiple instance learning differs from the traditional scenario in the way learning examples are encoded and represented. In the traditional supervised learning scenario, as mentioned earlier, each example is represented by one feature vector of fixed length. However, in MIL, the classifier must be designed and trained on a set of bags instead of feature vectors. Each bag consists of a collection of feature vectors called instances. Each such bag is labeled, but the label of the individual instances are unknown. In fact, instance labels can be indirectly inferred from the bag labels. MIL problems are often considered to be two-class problems, i.e., a bag can belong either to the positive or the negative class. In the standard MIL setting, a bag is labeled positive if at least one of its instances is positive, and a negatively labeled bag contains all negative instances. An MIL classifier seeks an optimal labeling scheme for unknown bags based on the above information.

More formally, in MIL, the training set is given by $D = \{< B_i, y_i >\}_{i=1}^{m}$, where bag $B_i = \{x_{i,j}\}_{j=1}^{|B_i|}$, $x_{i,j} \in \mathbb{R}^d$ is an instance and $|B_i|$ is the number of instances in B_i. Let $y_{i,j} \in \{0, 1\}$ be the latent variable of instance $x_{i,j} \in B_i$; then the label of B_i is known as $y_i = max\{y_{i,j}\}_{j=1}^{|B_i|}$. In other words, $y_i = 1$ if and only if at least one $x_{i,j}$ in B_i is a positive instance of the underlying concept; otherwise, $y_i = 0$. The training algorithm then automatically explores instance-level and bag-level models to find the one that best fits the given bag

labels and generalizes it to predict the labels of new bags or new instances [28].

The MIL problem was first proposed by Dietterich et al. [28] to solve the drug activity prediction. Ever since, it has increasingly been applied to a wide variety of tasks. In fact, many problems in computer vision and machine learning can be naturally cast in an MIL setting. The applications include but not limited to drug discovery [29], text classification [30, 31], computer-aided medical diagnosis [32], segmentation [33], image annotation [34], visual tracking [35,36], human detection [37], image classification [38–40] and content-based image retrieval (CBIR) [41–43].

In the context of image classification, MIL is needed when images are coarseley labeled at the image level and not at the region level. In such setting, an image is represented by a bag of instances where each instance corresponds to a feature vector extracted from different regions within the image. For instance, an image labeled as *vehicle* would contain at least an instance representing a vehicle, whereas images with other labels would not depict any vehicles-related region. Thus, a positive bag would include at least one instance that represents one of the image categories under consideration. The MIL approach attempts to use the labeled bags to model the concept object of interest in the instance space. Basically, in all of the image-based applications, the goal is to learn concepts from partially labeled images that can characterize each class. For instance, in basic image classification [38, 44] each image contains many objects, but only those representing information about target object/category are of interest. In other words, a positive instance can either be representative of a target object in the scene or the context of target class. Other instances may be shared across different classes and therefore have no discriminative information. To tackle the issue of intra-class variability, using MIL, we treat each image as a set of patches, but only those regions/instances that potentially carry category-specific information will be considered for classification.

1.3.2 Deep Learning

Over many years, image classification in computer vision has been relying on hand-crafted features such as SIFT [45], HOG [46] and Fisher vector (FV) [47] in conjunction with shallow discriminatively trained models. However, these features can only capture low-level edge information. The design of features to effectively capture mid-level information such as edge intersections or high-level representation like object parts becomes much more difficult.

In contrast with shallow learning algorithms, deep learning aims to extract hierarchical representations from large-scale data by using deep architecture models with multiple layers of non-linear transformations. With such feature representations, instead of raw pixel values or hand-crafted features, a better performance becomes more achievable. A deep neural network (DNN) is simply a feed-forward, artificial neural network that has more than one layer of hidden units between its inputs and outputs. The principle behind their success, however, is that they attempt to automatically unfold hierarchies of abstraction embedded in observed data in both unsupervised and supervised manners, by elaborately designing the layers depth and width, and accordingly selecting features that are beneficial for the learning task [48].

Deep learning architectures have different variants such as Convolutional Neural Networks (CNN) [49], Deep Belief Networks (DBN) [50], Deep Boltzmann Machines (DBM) [51], etc. Among them, CNN has been the most attarctive model which has been employed on large labelled datasets such as ImageNet [52], and has produced the best results on the most challenging image classification and detection datasets [53] by a huge margin. Additionally, CNN learns powerful generic image representations [54, 55] which can be used off-the-shelf to solve many visual recognition problems [55]. In contrast to hand-designed features used in previous methods, CNN is able to extract different levels of complex visual features from large-scale dataset using its multi-layer feed-forward structure. This approach provides a certain degree of simplification for Big Data analytics tasks, especially for analyzing massive volumes of data and discriminative tasks performing classification and

recognition. Most impressively, these approaches are capable of producing state of the art performance on tasks that the model was not explicitly trained for. This good generalization in performance on new tasks and datasets indicates that CNNs may provide a general and universal visual feature learning framework applicable to all tasks.

There are still some shortcomings in current deep learning practices. First, while we know that bigger models offer more potential capacity, in practice, the learned model is often limited by either too little training data or very limited time for running experiments, which can lead both to overfitting or underfitting. In essence, deep learning shares other machine learning methods's tendency to overlearn the training data. This means that the algorithm memorizes characteristics of the training data that may or may not generalize to the production environment where the model will be used. However, as mentioned earlier, this problem is not unique to deep learning, and there are ways to avoid it through independent validation.

Training deep architectures is difficult because the large number of parameters to be tuned necessitates an enormous amount of labeled training data that is often unavailable. Thereby, they require a great deal of computing power to build. While the cost of computing has declined dramatically, computing is still not free. For simpler problems with small data sets, deep learning may not produce sufficient added benefit over simpler methods to justify the cost and time. In other words, having enough amount of labelled training data, is essential to learning with CNN.

In this dissertation, we attempt to focus on the feature learning advantages of deep networks and employ it in our multiple instance learning framework.

1.4 Contributions

In this dissertation we address the VMMR application as an example of within-category object class recognition. VMMR presents a more diverse and challenging set of issues than in other fine-grained image classification problems (Refer to section 1.2.1). Thus, we need to focus on finer details of each class and instead of considering the whole

image/object, we use more distinctive patches/parts. In such scenario, we definitely need more than one visually discriminative region per class to be able to identify both the make and model. Our proposed approach will address both feature learning and part discovery simultaneously.

To tackle the above-mentioned challenges and issues, we propose and investigate an MIL-based algorithm for VMMR where each sample image is represented by a bag of instances. Let us suppose that we have assembled a training dataset with images labeled as positive if they contain *Toyota Camry*, and negative otherwise. Using an MIL approach, we can regard each training image weakly labelled with data as a bag containing a set of instances. The instances in a particular bag are various sub images each corresponding to a region of interest (ROI). If a bag is labeled as *Toyota Camry*, we know that at least one of the ROIs contains part of the vehicle. If a bag is labeled as non-*Toyota Camry*, we know that none of the sub images contain a vehicle with that make and model. In other words, the regions of interest (i.e. possible locations of the distinctive vehicle parts) in each image are considered as positive instances, and the rest are as negative instances. These regions could be obtained by segmenting the image into homogenous regions or by simply dividing the image into fixed-size blocks. Clearly, the data is ambiguously labeled (i.e. labels are available only at the bag level, and individual patches representing the vehicle are not labeled). We could just follow most existing MIL approaches and divide the image into n regions, and label all of them as positive if in the positive class. This way, most of these positive regions may belong to the background or non-discriminative parts of the vehicle which would result in many ambiguities in MIL. Additionally, using a very large dataset, we will be confronted with a very large feature space that could result in considerable computational overload. In this study, we focus only on a few ROIs as instances. In particular, we use saliency detection to capture more discriminative regions of the object as potential ROIs while limiting the number of instances.

Furthermore, we investigate the integration of CNN features in our proposed MIL framework and compare its performance with hand-crafted features. Specifically, we rely

on CNNs to learn appearance descriptors. By learning the features that are appropriate to describe the object categories in question, we let the data determine which features are effective for discrimination, which helps avoid losing information useful for categorization. By keeping part discovery completely unsupervised with respect to part annotations, we aim to make our algorithm scalable to a variety of fine-grained domains, including ones for which it is not known a priori which parts are discriminative.

The considered VMMR application requires a large and diverse dataset. To validate the proposed approach appropriately, we have created a very large dataset that includes images that were taken by different users, assorted imaging devices, and multiple view angles, ensuring a wide range of variations to account for various scenarios that could be encountered during testing. The data covers most makes and models that were manufactured after 1950.

The major contributions of this dissertation are summarized as follows:

1. We propose and evaluate unexplored representation and classification approaches for VMMR, based on the MIL and Deep Learning paradigms, and prove their effectiveness in realistic scenarios.

2. We develop an unsupervised method to select instances based on intrinsic features extracted from different regions. The instances are chosen regardless of the prior class information or viewpoint changes. This process leads to maximizing the number of relevant instances while minimizing the number of irrelevant negative instances in the process of multiple-instance learning.

3. We address an essential question of MIL: which instances indeed contribute to the semantic meaning of the bag-level labels?

4. To learn the key characteristics and features from all classes of makes and models in an optimised manner, we evaluate the potency of CNN features compared to hand-designed features.

5. The proposed VMMR approaches are compared to state of the art methods and are

evaluated using a comprehensive dataset that we have collected. This VMMR dataset is the largest public vehicle dataset available.

6. We evaluate our VMMR system on a real-life scenario, in a traffic multiple camera network. The unique challenge in traffic-image based MMR is due to various image qualities, having multiple vehicles, occlusion and the change in illuminance conditions and viewpoints.

1.5 Thesis Outline

The remainder chapters of this dissertation are structured as follows.

- Chapter 2 provides a review of multiple instance learning, instance selection and regional proposal techniques. It also, presents an extensive literature review showing how representative works in vehicle identification and classification have evolved over the years. Additionally, it illustrates the concept of deep learning and lays out different successful architectures in the context of content-based image understanding.

- Chapter 3 introduces our end-to-end multiple instance pipeline.

- Chapter 4 presents our second MMR scheme using a multiple instance CNN-based learning framework.

- Chapter 5 evaluates the performance and efficiency of the proposed VMMR approaches on existing benchmarks. It, also, presents the experimental results and analysis on the selected discriminative instances.

- Finally, chapter 6 concludes by summarizing the work, and outlines possible future directions of research.

CHAPTER 2

BACKGROUND

In this chapter, we review existing approaches in areas that are highly relevant to our proposed research. First, we outline the Single Instance Learning problem in the context of fine-grained recognition. Then, we describe the Multiple Instance Learning paradigm and illustrate its advantages over Single Instance Learning for certain tasks. Next, we provide literature review on region selection methods with a focus on saliency detection techniques. Then, we outline the concept of deep learning and lay out different successful architectures in the context of content-based image understanding. Finally, we review the literature in the area of Vehicle Make and Model Recognition.

2.1 Single Instance Learning for Fine-grained Classification

Fine-grained recognition is the task of discriminating between similar objects with subtle differences. A straightforward idea to solve such problems is to simply apply methods used for generic classification tasks regardless of the granularity of the class differences. However, classification algorithms for basic-level tasks and fine-grained-level tasks are quite different.

For basic level classification, the difference between classes is significant and can be computed by one global feature vector. For instance, the Bag-of-Words model [56] has been used frequently to accumulate features extracted from different parts of the object. This strategy does not scale well with respect to the number of classes and the number of subordinate classes in fine-grained classification problems is usually very large. In fact, subtle differences between classes could be missed because of the relatively large number of similar features. Thus, for fine-grained classification, commonly used methods focus on extracting

more detailed features from various parts of the image to improve the chance of capturing information that discriminate between classes [57].

For supervised learning, class labels are needed and are used to learn the recognition models using statistical machine learning techniques. First, features that capture intrinsic properties of the objects are extracted. Then, a decision function is learned to map the constructed descriptors to their category labels. The learned decision function is then used to predict the label of a novel test object.

Let x_i be the feature vector extracted from object o_i. x_i lies in some input space $\mathbb{R}^d$ and has a label y_i in a label space $\mathbb{Y}$. Given the the training data $D = \{< x_i, y_i >\}_{i=1}^{n}$ with n samples, drawn from a (unknown) distribution $\mathcal{P}(x, y)$, the goal of the classifier is to learn a hypothesis $g : \mathbb{R}^d \rightarrow \mathbb{Y}$, such that $g(x)$ approximates y for new samples $(x', y') \sim \mathcal{P}$. The first decision to make pertains to the hypothesis class (or model) $\mathcal{G}$ from which to select g. A wide range of options are available and the best choice often depends on assumptions about the data (linear separability, presence of outliers, amount of data, prior knowledge). If $\mathcal{G}$ is too large, the model might overfit the training data. On the other hand, if $\mathcal{G}$ is too small the model will underfit and may result in poor generalization.

In fine-grained classification approaches, features are typically extracted from different parts of the object in an attempt to capture more detailed and potentially more discriminative information. Object parts are either manually identified and annotated [58,59], or discovered by a set of object detectors [4]. For the former, the annotation quality is good but the annotation process does not scale with the number of images in the training dataset and with the number of classes. This process remains challenging even when the labeling task is crowd-sourced. Ideally, members of the crowd should be able to recognize and accurately annotate the objects in question. However, in reality workers have different levels of competency and attention. Moreover, while these selected parts are guaranteed to be human-understandable (which suffices for human-in-the-loop classification applications [60]), they may not be machine-detectable and hence may not work well in automated systems.

For the automated object detection approach, an accurate registration is almost impossible because of the variations of the image structures across different classes. Also, a local attribute might correspond to features at different unknown positions and scales across images. Usually, a set of object component detectors can be pre-defined for a specific domain. This can make the approach more reliable. However, it makes it more difficult to adopt to other domain or even other datasets. Missing parts for registration is another obstacle of these approaches. Therefore, there is need to develop new algorithms for fine-grained tasks to make them suitable for wide deployments. The aforementioned problems can be alleviated using weakly supervised localization which can learn with partially labeled data.

2.2 Multiple Instance Learning

Multiple Instance Learning (MIL) is a supervised learning framework that aims at handling ambiguously labeled data in classification and regression problems [61]. As opposed to traditional supervised learning, where the learning procedure works over fixed-length vector of features as instances and their corresponding labels, MIL operates over bags of instances, where each bag is composed of multiple instances. Each bag can contain a different number of instances. This form of learning is referred to as weakly supervised, since the labels of bags are known but not those of individual instance. The key assumption of MIL is that a bag is labeled as positive, if at least one of the instances within the bag is known to be positive, whereas it is labeled as negative, if all the instances are known to be negative. Positive bags can encode ambiguity since the instances themselves are not labeled. Given a training set of labeled bags, the goal of MIL is to learn a concept that predicts the labels of training data and generalizes to predict the labels of testing bags [28]. Figure 2.1 depicts the differences between these paradigms of learning.

Existing methods in solving MIL problem fall into two categories, generative models and discriminative models.

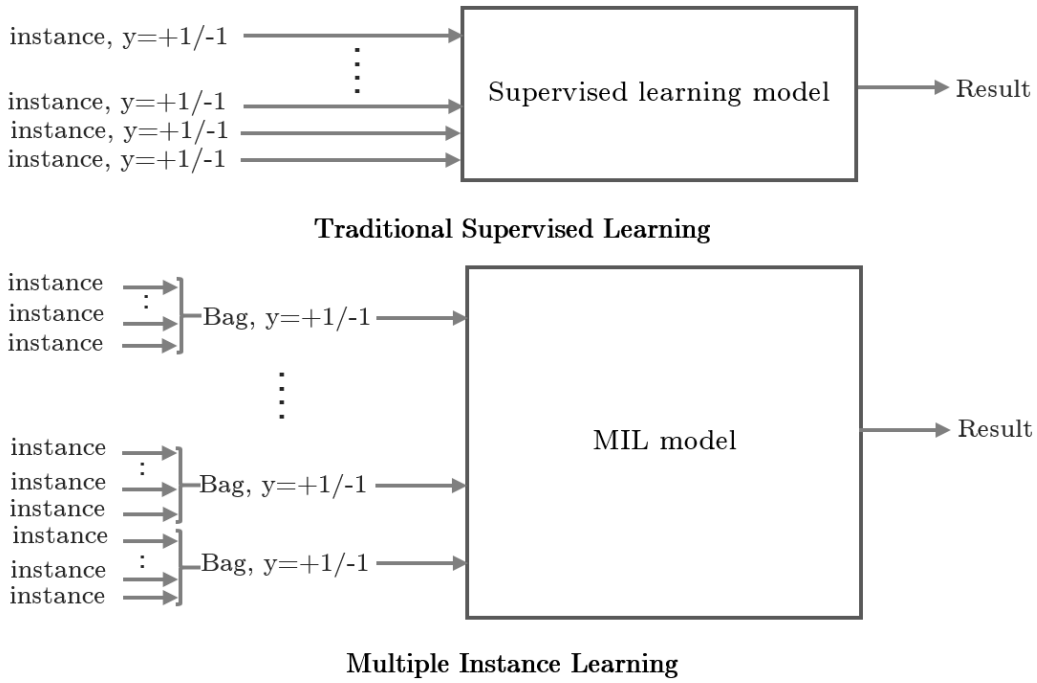


Figure 2.1: Differences between traditional supervised learning and multiple instance learning for the case of two-class problem. A representation based on [28]

2.2.1 Generative Models for MIL

Generative model-based algorithms attempt to learn a single target distribution to generate instances and/or bags and their labels in a joint manner. They represent the target concept by a region in the instance space which covers as many instances from positive bags as possible while excluding as many instances from negative bags as possible. Axis parallel hyper-rectangle [28], Diverse Density (DD) [29] and Expectation Maximization DD (EM-DD) [62] are examples of algorithms that fall into this category. In many generative algorithms, the bag label is predicted by first estimating the hidden labels of its instances.

2.2.1.1 Axis-parallel Rectangle

The axis-parallel rectangle (APR) method [28] attempts to find a hyper-rectangle in the feature space to represent the area of the true positive instances. Intuitively, this rectangle is identified as a region in the instance feature space that includes at least one instance from each positive bag and excludes all instances from the negative bags.

Dietterich et al. [28] suggested three algorithms to find such a hyper-rectangle; a "standard" algorithm finds the smallest APR that bounds all the instances from positive bags; an "outside-in" algorithm constructs the smallest APR that bounds all the instances in positive bags and then shrink the APR to exclude false positives; an "inside-out" algorithm starts from a seed point and then grows a rectangle from it with the goal of finding the smallest APR that covers at least one instance per positive bag and no instances from negative bags. These algorithms were employed to solve the problem of drug activity prediction, where the potency of a drug is determined by its binding degree with a target molecule and binding strength of a drug is determined by the shape of the drug molecules. This is a MIL problem since the molecules may adopt many possible shapes by simple rotation of internal bonds. A drug is classified as *active* if at least one of its instances is inside the APR, otherwise, it is classified as *inactive* [28].

Although APR works well for some problems, in other applications, such as image classification, it is quite possible that no APR could be found to satisfy the criteria. Moreover, APR is very sensitive to labeling noise. For instance, having only one mislabeled negative bag would force the algorithm to include at least one instance from this bag. This would result in an APR that contains many negative instances and it cannot represent the area of true positive instances [63].

2.2.1.2 Diverse Density

The Diverse Density (DD) algorithm is another common MI learning framework. In [29], Maron and Lozano-Perez proposed the DD measure with the objective to find a "soft" region that describes the intersection of the positive bags minus the union of the negative bags. To achieve this, DD uses a gradient search algorithm, with multiple starting points selected from a set of instances in positive bags, to find the target concept. The target concept is defined as a point in the instance feature space that is "close" to at least one instance from every positive bag but "far" from instances in the negative bags. In other words, the target concept describes a region that is densely populated by instances from

positive bags. Thus, the task of MIL is transformed to the search for this prototype point in the feature space which holds the maximal DD value.

Formally, let $\{B_i^+\}_{i=1}^n$ and $\{B_i^-\}_{i=1}^m$ denote the n positive and m negative bags in the training dataset respectively. The diverse density of a given target concept t is defined as the probability that t is the correct concept over the feature space.

$$DD(t) = Pr(t \mid B_1^+, B_2^+, \ldots, B_n^+, B_1^-, B_2^-, \ldots, B_m^-) \tag{2.1}$$

Using Bayes rule and assuming uniform prior over the target concept location, this is equivalent to maximizing the following likelihood:

$$DD(t) = Pr(\ B_1^+, B_2^+, \ldots, B_n^+, B_1^-, B_2^-, \ldots, B_m^- | t\ ) \tag{2.2}$$

Under the assumption that all bags are conditionally independent, given the concept point t, (2.2) can be decomposed into:

$$DD(t) = \prod_{1 \le i \le n} Pr(B_i^+ \mid t) \prod_{1 \le i \le m} Pr(B_i^- \mid t) \tag{2.3}$$

Using Bayes' rule further and assuming uniform priors, Maron and Lozano-Perez [29] showed that optimizing DD in (2.3) is equivalent to optimizing $\widehat{DD}$, defined as:

$$\widehat{DD}(t) = \prod_{1 \le i \le n} Pr(t \mid B_i^+) \prod_{1 \le i \le m} Pr(t \mid B_i^-) \tag{2.4}$$

In (2.4), Given the fact that boolean label of a bag is the result of "logical-OR" of the labels of its instances, $Pr(t \mid B_i)$ is defined using the *noisy-or* model [64], i.e.

$$\begin{cases} Pr(t \mid B_i^+) & = 1 - \prod_j (1 - Pr(t \mid B_{i,j}^+)), \\ Pr(t \mid B_i^-) & = \prod_j (1 - Pr(t \mid B_{i,j}^-)) \end{cases} \tag{2.5}$$

where $Pr(t \mid B_{i,j}^+)$ (or $Pr(t \mid B_{i,j}^-)$) can be estimated by a Gaussian-like distribution

$$Pr(t \mid B_{i,j}^+) = exp(-\|B_{i,j}^+ - t\|^2) = exp(-\sum_k w_k (B_{i,j}^+ - t_k)^2) \tag{2.6}$$

where w_k is a non-negative scaling factor that reflects the degree of relevance of features. The target concept t can be found by maximizing (2.4).

The DD method has been widely used for different MIL applications such as drug activity estimation [29] and image retrieval [65]. However, it is sensitive to labeling noise.

The EM-DD [62] is another algorithm that seeks an optimal target concept. It uses the Expectation Maximization (EM) algorithm to maximize $\widehat{DD}(t)$ in (2.4).

In the MIL definition, the label of a bag is determined by the "most positive" instance in the bag. The problem, however, comes from the ambiguity of not knowing which instance is the most likely one. The EM-DD attempts to model the instance labels using hidden variables. It starts by taking an initial guess from positive instances as a target concept t (which can be obtained using DD algorithm). Then it alternates between two steps: in the E-step, the current hypothesis of concept t is used to pick one instance from each bag which is most likely responsible for the bag label; in the M-step, it finds a new target concept t' by maximizing the likelihood over all negative and positive instances identified in the E-step. These steps are repeated until the algorithm converges.

EM-DD turns the multi-instance problem to a single-instance definition by removing the noisy-or part of the DD algorithm. This modification, highly reduces the complexity of the optimization function and computational time.

The above methods are quite efficient in learning, but they are based on the assumption that all positive instances form a tight cluster in the feature space [30]. It means that they consider mainly the information of only one or a limited number of prototypes to represent the target concepts in the positive bags. This kind of assumption might not be necessarily true for many real-life applications with diversified positive instances. There are, in fact, more than one instances from the concept in many positive bags and much of the information contained in these instances is lost this way. Moreover, the true positives may not follow a Gaussian distribution. Indeed, the distribution of positive instances can be multi-modal or even random, and the single target distribution learnt by these methods may fail to cover the whole instance space [39].

To tackle this problem, in [66], Karem and Frigui proposed a method to identify multiple target concepts simultaneously. They addressed the ambiguities arising from not having

any information on the relevance of each feature and ending up with only a few relevant instances, they employed a fuzzy approach. They extended the DD model using a fuzzy clustering approach to introduce a fuzzy Multi-target concept Diverse Density (MDD) metric, assuming that each bag B_i belongs to each target concept t_k with a membership degree u_{ki}.

$$MDD(T,U) = \prod_{i=1}^{N} \prod_{k=1}^{K} u_{ki}^{m} Pr(t_k|B_i) \tag{2.7}$$

where $U = [u_{ki}]$ for $k = 1, \ldots, T$, $i = 1, \ldots, N$, and m is a fuzzifier that controls the fuziness of each partition. The MDD is maximized when the target concepts correspond to dense regions in the feature space with maximal correlation to instances from positive samples, and minimal correlation to instances from negative samples.

2.2.2 Discriminative Models for MIL

Discriminative model-based methods, including DD-SVM [67], MI-SVM [68], MI-Kernel [69], MIO [70], MILES [38] and Citation-kNN [71], focus on modeling data and/or bag labels given features of data instances or bags. Some of these methods extend standard SIL approaches to the MIL setting. Other methods map bag features into a simple high dimensional feature vector, then, apply standard classifiers. Using many benchmark datasets, it has been shown that these discriminative methods tend to be more robust and accurate than generative algorithms [72].

As mentioned earlier, using the standard MI assumption, if a bag contains at least one positive instance it is labeled as positive, otherwise, it is labeled as negative. This assumption is suitable for some problems such as drug activity estimation [28]. This is because one of the molecules shapes is sufficient to infer if the drug has potency. However, recent work has applied MIL to other domains that require alternative MIL assumptions. Consequently, new MIL techniques with more relaxed assumptions have emerged. For instance, in image classification applications, negative bags may contain parts of positive category instances.

Chen and Wang [67] proposed a MIL framework named DD-SVM, where a bag label is

not determined by the standard assumption, instead by the number of instances satisfying certain properties. In their method, initially, using the DD function, a set of instances are selected. These instances are more likely to appear in positive bags than negative ones. Then, based on the feature space constructed from a mapping defined by the local maximizers and minimizers of the DD function, a Support Vector Machine (SVM) in combination with radial basis function (RBF), is trained to discriminate between positive and negative bags. In essence, DD-SVM converts MIL to a standard supervised learning problem through feature mapping. Ray et al. [30] extended the DD framework by using a Logistic Regression algorithm to estimate the equivalent probability for an instance. They employed a soft max function to combine instance-level information to predict the bag label. Weidmann et al. [73] considered a generalization where the presence of a combination of instance types determines the label of the group. Xu and Frank [74] assumed that all instances contribute equally and independently to a group's class label. These types of solutions are typically application dependent and tailored to handle specific assumptions about the whole-part relationship between groups and instances.

2.2.2.1 MI-SVM

Andrews et al. [68] proposed mi-SVM and MI-SVM methods as modifications to the standard SVM for instance-level and bag-level classification, respectively. Both of these approaches, formulate the learning problem as a mixed integer problem and attempt to maximize the margin of the instance classifier.

The objective of mi-SVM is to maximize the instance margin jointly over unknown hidden instance labels and the kernel parameters. Hence the same formulation is used as SVM, but the minimization is done over the individual labels as well, subject to the constraint that in a positive bag, at least one instant should have a positive label and all instances should have negative labels in a negative bag. In mi-SVM, instance labels y_i are considered as unobserved hidden variables subject to constraints imposed by their bag labels Y_I. In case of a linear discriminant function $f(x) = w^T x + b$, the weight vector $w \in \mathbb{R}^d$ and offset

$b \in \mathbb{R}$ are estimated to find integer values $y_i \in \{+1, -1\}$:

$$\begin{aligned} &\min_{\{y_i\}} \min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_i \xi_i, \\ &\text{s.t.} \quad \forall i : y_i(< w, x_i > +b) >= 1 - \xi_i, \quad \xi_i >= 0 \end{aligned} \tag{2.8}$$

In (2.8), C is a regularization parameter and ξ is the slack variable.

Inferring the latent labels and training jointly the instance classifier is a hard mixed integer problem that is typically solved by alternating optimization. It consists of two main steps: inferring the labels, and classifier learning. Given the discriminant function, the algorithm finds the integer variables y_i that correspond to the unknown instance labels in the training bags. Then, given the inferred instance labels from the previous step, it finds the optimal parameters (w, b) of the discriminant function. These two steps are performed on the same training bags sequentially, which means that the same instances are used for both training the discriminant function and assigning the missing labels. However, inferring the latent labels with the classifier which has been evaluated on the same data as the one it was trained on makes the MIL procedure very susceptible to overfitting.

In comparison, the objective of MI-SVM is to maximize the bag margin, where the bag margin is defined by the margin of the *most positive* instance of each positive bag, or the margin of the *least negative* instance in case of negative bags [68]. The algorithm solves the following non-convex optimization problem:

$$\begin{aligned} &\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_I \xi_I, \\ &\text{s.t.} \quad \forall I : Y_I \max_{i \in I}(< w, x_i > +b) >= 1 - \xi_I, \quad \xi_I >= 0 \end{aligned} \tag{2.9}$$

MIL-SVM learns in two loops: the outer loop sets the values of the selector variables by assigning the average of all instances feature values in that bag for initialization. The inner loop trains a standard SVM, in which the variables are tuned and the selected positive instances are used to replace the positive bags. The re-labeling process continues until selected instances for bag representation do not change significantly.

In the mi-SVM algorithm, the margin of every instance matters. In MI-SVM, however, only one instance per bag matters since it determines the margin of the bag. The former

is suitable for tasks where individual instance labels are important, while the latter is used for tasks where only the bag labels are concerned.

2.2.2.2 MILES

MILES [38] converted the MIL problem into a standard supervised single instance learning problem by mapping each bag into a feature space defined by the similarity between its instances and a set of target concepts. Unlike most earlier DD-based methods, MILES assumes that multiple target concepts may exist, and therefore assigns a probability to each instance to represent a target concept given a bag. Formally, for a given bag B_i of instances $x_{i,j}$, $j = 1, \ldots, m$, its similarity to a given instance x, is given by:

$$Pr(x|B_i) \propto s(x, B_i) = \max_j \Big\{ exp\Big(- \frac{||x_{i,j} - x||^2}{\sigma^2} \Big) \Big\}, \tag{2.10}$$

where σ is a scaling factor.

In other words, each instance in the training bags can be a candidate for a target concept. The candidates are represented as features in an instance-space feature space. Using (2.10), each bag in the training set is mapped into this space induced by the similarity values. i.e. a bag is represented by the coordinates $m(B_i)$ as following:

$$m(B_i) = [s(x_1, B_i), s(x_2, B_i), \ldots, s(x_C, B_i)] \tag{2.11}$$

where $x_i \in C$ is an instance from the set of all instances in the training bags.

Considering a binary MIL classification problem, with bag labels of $+1$ and -1, MILES uses the 1-Norm SVM [75] to learn a linear classifier on the mapped space and perform instance selection. i.e.,

$$y_i = sign(\sum_{k=1}^{C} w_k s(x, B_i) + b) \tag{2.12}$$

where w_k is a weight associated with $s(x, B_i)$ and b a bias parameter.

2.2.2.3 Citation k-Nearest Neighbors

In the standard k-Nearest Neighbors (k-NN) classifier, to classify a given instance, the k nearest instances are retrieved using a distance measure on the instance space (e.g.

Euclidian distance), then an output label is computed from the labels of the k nearest instances. Using the same approach, Wang and Zucker [71] adapted k-NN for the case of having multiple instances. To determine the nearest neighbors for a given bag, they used the Hausdorff metric (2.13) instead of the Euclidian distance. They found that the majority vote method, used by the standard k-NN, often produced sub-optimal results in the multiple instance setting, since it would easily get confused by the false positive instances in positive bags [76]. To improve the multiple instance k-NN, they proposed a variation called Citation-kNN [76], where a bag is labeled through analyzing not only its neighboring bags (known as references) but also the bags that regard the bag being tested as a neighbor (known as citers).

Formally, given two bags B_1 and B_2 of instances $\{x_{1,j}\}_{j=1}^{m_1}$ and $\{x_{2,j}\}_{j=1}^{m_2}$, respectively, the Hausdorff metric between B_1 and B_2 is defined as

$$\mathbf{H}(B_1, B_2) = \min_{x_{1,j} \in B_1, x_{2,j} \in B_2} \left\{ dist(x_{1,j}, x_{2,j}) \right\} \tag{2.13}$$

where *dist* is a distance measure between instances (e.g. Euclidian distance).

Citation-kNN is motivated by the notion of citation from library and information science. Under this view, the authors defined a *C-nearest citers* measure for a given bag. This measure is defined as following:

- For two given bags, B and B', let $Rank(B', B)$ equal n if B is the nth nearest neighbor of B'.

- Then, the *C-nearest citers* of B are the C bags that return the lowest neighbor ranking for B. i.e.,

$$Citers(B, C) = \{B_i \mid Rank(B_i, B) \leq C,\ B_i \in \mathcal{B}\} \tag{2.14}$$

  where $\mathcal{B}$ is the set of all training bags.

The decision of Citation-kNN relies on the K-nearest bags (references) as well as the C-nearest citers. Specifically, a bag is classified as positive if and only if there are strictly more positive bags than negative bags in the combined K-nearest bags and C-nearest citers. C is

usually set to K+2. It should be noted, however, that unlike the DD method, Citation-kNN is unable to predict the labels of instances.

2.2.3 Instance Selection for MIL

In MIL, each object is represented by a bag of instances. Negative bags should include only negative instances and positive bags should contain at least one positive instance. Thus, in addition to learning the classifier, the selection and representation of instances plays an important role in MIL. For instance, in image classification an image can be represented as a bag of smaller image patches. Several methods have been used to select the instances of an image. For example, an image can simply be represented by a collection of fixed-size blocks [43, 65] or even regions obtained from image segmentation [67]. During training, MIL would find those patches that lead to best classification results and leave out the others.

However, not all image regions are related to the object/subject of interest. Another problem is that in the process of image patch generation, many discriminative regions might split over multiple instances. Also, an image patch might be too large that it contains multiple targets of concept. All these issue would cause the positive instances to lose their discriminative properties. Additionally, the number of instances per bag should be large enough to make sure that it capture all the discriminative regions in the image. Thus, another potential problem that hinders the efficiency of MIL is the possible large number of instances encountered in real-world applications. In other words, the high-dimensional instance space results in high complexity for both the feature computation and the classification process.

To overcome these limitations, alternative modeling assumptions have been adopted for MIL algorithms. For example, a bag's label is determined by a classifier taking into account part of its instances [31, 77]. DD-SVM [67] learns the multiple instance prototypes using the start values defined by local extrema estimated by the EM-DD cost function within each training bag. The algorithm's performance is highly affected by the labeling noise since a negative bag close to a positive instance reduces the DD value of the instance considerably,

and as a result its chance to be selected as a prototype. The other problem, is that EM-DD optimization should be applied to all training instances to find the local extrema. MILES [38], on the other hand, as a more efficient approach, does not explicitly select the instance prototypes. It maps every bag into an instance space whose dimensionality is given by the total number of instances across all bags. Having the new feature map, it employs the 1-Norm SVM [75] to select concepts and train classifiers. This will result in the main drawback of MILES which is a high-dimensional bag-level feature vector which leads to high complexity for both bag mapping and SVM optimization, and containing too many irrelevant features [38]. Although these methods are very robust and achieve a higher performance than the aforementioned category, they are considerably time consuming for large datasets.

To address this problem and achieve acceptable performance with much less complexity, we need to refine the instances and reduce the false positives as much as possible. It is also important that instance selection, as a preprocessing step, should be done efficiently especially for large-scale datasets. Different perspectives on this issue has led to different MIL approaches.

MILD [63] proposes an instance selection mechanism based on a conditional probability model developed to identify the true positive instance in a positive bag. To achieve this goal, a decision function is formulated whose accuracy on predicting the labels of the training bags is used to measure true positiveness of the corresponding instance. MILIS [44] addresses the high-dimensionality issue by initially selecting an instance prototype from each bag and all the selected prototypes are iteratively optimized by an EM-like framework. A kernel density estimator is first learned from all the negative instances in negative bags to reduce the number of positive candidates. Based on the distribution estimate, the most positive (i.e. the least negative) instance per positive bag is selected to represent the concept. Linear SVM is then applied to train the classifier based on the feature space for the bag-level embedding constructed by these instance prototypes. Although it is much more efficient, MILIS becomes vulnerable to labeling noise and often learns too few concepts [78].

2.3 Region Proposal

In the context of image classification, the global description of a whole image is too coarse to achieve good classification and retrieval accuracy. Thus, to fit this problem into the MIL setting, each labeled image can be a bag of pixels/patches which are modeled as instances. A label is assigned to the whole image aggregating the information gathered from instance level [44], based on which the target concept representing the target object can be learned through MIL algorithms. Loosely speaking, if an image patch repeatedly occurs in different positive bags, it would be called a concept. Then, the weight of each concept is an indicator of its frequency in positive bags.

To illustrate, consider the task of classifying images that contain *vehicle*. In this application, given an input image we want to determine whether it contains a vehicle. Using an MIL approach, each instance corresponds to a ROI. These regions could be obtained by segmenting the image into homogenous regions or by simply dividing the image into blocks. A multiple instance representation is well suited for this purpose because only few regions may contain the object representing the target vehicle. This representation is illustrated in Figure 2.2.

Figure 2.2: Example of an image represented as a bag of 24 instances.

Inducing concepts correctly is a very important part of MIL algorithms. But, the hidden nature of the instance labels poses great challenges for MIL. As mentioned in section 2.2.3, depending on the relative size of the object of interest, many of these regions may be background or not representative of the object of interest, or the segmentation methods

may cut the object of interest into multiple components, all resulting in many irrelevant instances to the target class.

A default assumption in many MIL algorithms is that a positive bag must contain at least one true positive instance, whereas a negative bag contains negative instances only. Due to subjective labeling, existence of image distortions such as changes of scale, or illumination and view angle, some positive bags may not contain true positive instances. Similarly, some true positive instances could be included in some negative bags [78].

Additionally, we will be confronted with too many instances for even average-scale real-world datasets. Considering all possible patches in each positive image, an exhaustive search for object locations would have to consider a very large number of hypotheses [44].

Furthermore, in many cases, negative instances extracted (from patches that correspond to background) may be even more persistent in positive bags, which leads to a decrease in performance [44].

One approach towards alleviating these issues is to design efficient instance pruning techniques to speed up the training process without compromising the performance. Some methods have attempted to decrease the complexity by sampling the patches in positive images. Crandall et al. [79] took a random sample of patch descriptors from training images and initialized the training with the most discriminative subset. Several part-based models were then initialized from descriptor pairs and subsequently optimized through EM. In [80], Chum et al. followed a clustering approach by starting from visual words of a BoW representation and proceeded with the MIL approach. Random initialization can also be avoided by filtering patch candidates in positive images. Deselaers et al. [81] applied a trained generic object detector to guide initialization of 100 random samples in each training image. By assuming that there is only one object in each positive image they trained a Conditional Random Field (CRF) which simultaneously optimizes object locations and the classification model.

Another approach is to initialize positive object locations to entire (or almost entire) positive images and then attempt to gradually zoom into correct locations through iteration.

Cinbis et al. [82] expressed this iteration in terms of bottom-up location proposals. In their proposed method, called multi-fold MIL learning, the first classification model is trained on Fisher vectors of entire positive and negative images. In each subsequent iteration the negative locations are chosen as (false) positives of the current classification model on the negative training dataset. On the other hand, the positive locations are set to the top-scored bottom-up location proposals. They avoided a bias towards the locations from the last iteration by performing the training and selection steps on different folds of the training set of positive images.

Some methods attempt to reduce the number of irrelevant instances and select subset of patches by choosing the ones invariant to certain geometric transformations [56, 83, 84]. Csurka et al. [56] proposed a bag-of-keypoints model for object classification, in which each bag was represented by a collection of affine invariant patches. A predetermined number of clusters was generated by quantizing descriptors of all image patches. Each image was then transformed to an integer-valued feature vector indicating the number of patches assigned to each cluster. CkNN-ROI [85] and KI-SVM [86] were proposed to locate regions of interest (ROI) in image analysis. In these approaches, the most positive instance in a bag would be identified as the best ROI.

2.3.1 Region Selection in Fine-grained Classification

In fine-grained image classification, many approaches have tackled this issue by representing parts of objects as instances. In [87], over-segmented regions in images were used as parts and linear discriminant analysis (LDA) was employed to learn the most discriminative ones for scene recognition. Gavves et al. [88] segmented images via GrabCut [89], and then roughly aligned objects by parameterizing them as an ellipse. In [90], discriminative parts were selected through the mean shift method on local patches in images for each class. All these methods attempt to evaluate each part, which may be very computationally expensive when the part number is very large.

Some works consider a more practical setup when part annotations are missing in the test-

ing phase. They learn part detectors from annotated parts in the training images and apply them on testing images to detect parts. For example, in [3], the poselet [91] was used to detect object parts. Then, each object was represented with a bag of poselets and the top matchings among poselets (parts) were found to match objects. Zhang et al. [92] used the deformable part models (DPM) [93] for object part detection. DPM was learned from the annotated object parts in training objects, which was then applied on testing objects to detect the head and body in birds. The candidate yielding the highest response to a certain part detector was used as the detected part in the object. In [94], Chai et al. performed a joint segmentation and DPM model fitting, extracting features around each DPM part. Some works [95] transfer the part annotations from objects in training images to those sharing similar shapes in testing images instead of applying object/part detectors.

Recently, [96] proposed to use object and part detectors with powerful Convolutional Neural Network (CNN) feature representations [97]. The outputs from the inner convolutional (CONV) layers can be seen as the feature representations of sub-regions in the image. When CNN is used on an object proposal, the outputs from the inner convolutional layers can be seen as the part representations. Simon and Rodner [98] first generated a pool of parts by using the outputs from all layers in CNN. Then, they selected useful ones for categorization. They considered two ways of selection: one was to randomly select some parts; the other was to select a compact set by considering the relationship among them. These parts were concatenated to represent the image. Jaderberg et al. [99] learned to detect and align objects in an end-to-end system. This system includes two parts: one is an object detector, which is followed by a spatial transformer. The spatial transformer is learned to align the detected objects automatically to make the parts match accurately. Recently, CNN aided by region proposal methods, has become popular in object recognition/detection, e.g., RCNN [100], faster-RCNN [101], and RCNNminus-R [102]. All these methods focus on the supervised object detection, where object bounding boxes in training images are necessary to learn the object detectors.

Most of the above approaches may completely miss small objects at the initialization step.

Due to that, MIL refinement may easily get trapped in a local optimum, and the training is likely to fail. Additionally, MIL optimization is computationally very intensive making training on large datasets not practical.

When compared to traditional sliding window based object detection paradigm [46, 93], however, estimating region proposals in a pre-processing stage has three major advantages: it better follows human mental recognition behavior which quickly perceives distinct regions before identifying objects [103]; by reducing the search locations (e.g., from typically a few millions to less than a few thousands), it speeds up the computation considerably, especially when the number of object classes that need to be detected is high; and finally it improves the detection accuracy by employing more discriminative features during classification [104].

One of the main objectives of this dissertation is to address the aforementioned issues and investigate methods to select more potential representative instances to improve the MIL accuracy while keeping the number of instances reasonable.

2.3.2 Saliency Detection

Saliency intuitively characterizes some parts of a scene, which could be objects or regions, that appear to an observer to stand out relative to their neighboring parts [105]. Visual saliency endeavors to identify these visually distinctive parts in an image. Humans are able to effortlessly and rapidly perform this task [106]. After filtering these regions, they perceive and process them in finer details to extract high-level information. Many computer vision applications have employed visual attention models that focuses on finding locations of images that capture early-stage human fixations to find the objects or regions that efficiently represent a scene; such as object recognition [107, 108], image segmentation [109, 110], video summarization [111, 112], content-aware image resizing [113] and ,content-based image retrieval and image collection browsing [114, 115].

As visual attention is specific to salient objects, all the fixations on the salient object are generally not restricted to a specific region. Instead, fixations tend to cluster

around regions of interest within the salient object. The effects of this mechanism can be represented by a saliency map that topographically records the level of visual attention priority. Cognitive scientists have been doing extensive research on this capability for a long time. Some studies, to contribute to perception, understanding, and representation of a scene, have focused more on the distinctive regions and suggest that they are mostly related to uniqueness, rarity, contrast [106, 116] and sudden changes, characterized by primitive features like color, texture, shape, etc. [117, 118]. Elazary and Itti [119], demonstrated that human annotators tend to consider more salient objects first. In [120] Masciocchi et al., using the mouse clicking data from a large observer population, found out that interest selections are correlated with eye movements, and both of them correlate with saliency.

Assume K subjects have viewed a set of N images $\mathcal{I} = \{I_i\}_{i=1}^{N}$. Let $L_i^k = \{p_{ij}^k, t_{ij}^k\}_{j=1}^{n_i^k}$ be the vector of eye fixation $p_{ij}^k = (x_{ij}^k, y_{ij}^k)$ and their corresponding occurrence time t_{ij}^k for the kth subject over image I_i; and let the number of fixations of this subject over the ith image be n_i^k. The goal of attention modeling is to find a function $f \in \mathcal{F}$ which minimizes the error on eye fixation prediction, i.e.,

$$\sum_{k=1}^{K} \sum_{i=1}^{N} m(f(I_i^k), L_i^k) \tag{2.15}$$

where $m \in \mathcal{M}$ is a distance measure [117].

Salient object detection can be performed either in a bottom-up fashion using low-level features based on characteristics of a visual scene, including gradient information and curvedness [121], local contrast and rarity features [122, 123], symmetry [124] and frequency-domain information [125], or in a top-down fashion via the incorporation of high level knowledge, expectations and task demands [113, 126, 127].

2.3.2.1 Bottom-up Approaches

To measure the saliency of regions, uniqueness, usually in the form of global/local regional contrast, is the most frequently used feature. In [128], a region-based saliency algorithm is introduced by measuring the global contrast between the target region with respect to all other regions in the image. The input image is first segmented into N regions

$\{r_i\}_{i=1}^{N}$. The saliency value of region r_i can be measured as:

$$s(r_i) = \sum_{j=1}^{N} w_{ij} D_r(r_i, r_j) \tag{2.16}$$

where $D_r(r_i, r_j)$ captures the appearance contrast between regions r_i and r_j. Regions with large global contrast will have higher saliency scores. In (2.16), w_{ij} is a weight term between two regions, which can serve as a spatial weighting by giving farther regions less contributions to the saliency score than close ones. In some cases, w_{ij} might be used to account for the irregular size of the target region depending on the segmentation approach.

Jiang et al. proposed a multi-scale local region contrast based approach [129], which calculates saliency values across multiple segmentations to boost robustness, and combines these regional saliency values $s(r_i^n)$ to get a saliency map for each pixel x:

$$s(x) = \sum_{n=1}^{N_s} \sum_{i=1}^{N(n)} s(r_i^n) c(x, r_i^n) \delta(x \in r_i^n) \tag{2.17}$$

where N_s is the number of different segmentation sets, $N(n)$ is the number of regions in the nth segmentation, and $c(x, r_i^n)$ is the color similarity of region r_i^n and its contained pixel x. As the salient object usually lies near the center of the image, they embedded a Gaussian weight in their saliency score calculations to emphasize regions around the image center. In addition to internsity/color contrast uniqueness, some methods have also employed other distinctiveness cues such as texture and structure to form the saliency map [130].

Some methods have considered spatial distribution prior to detecting the salient objects [131]. They start with the assumption that the wider a color is distributed in the image, the less likely a salient object contains this color. In such approaches the pixels in the input image, I, are quantized by a Gaussian Mixture Model (GMM) $\{\omega_c, \mu_c, \Sigma_c\}_{c=1}^{C}$, where $\{\omega_c, \mu_c, \Sigma_c\}$ are the weight, mean color and the covariance matrix of the cth component. Each pixel x is assigned to a color component with probability:

$$P(c|I_x) = \frac{\omega_c \mathcal{N}(I_x|\mu_c, \Sigma_c)}{\sum_c \omega_c \mathcal{N}(I_x|\mu_c, \Sigma_c)} \tag{2.18}$$

This way, the saliency of each pixel would be defined as:

$$s(x) = \sum_c P(c|I_x)(1 - V(c)).(1 - D(c)) \tag{2.19}$$

where $V(c)$ is the spatial variance of component c and $D(c)$ is a center-weighted normalization term to balance the border cropping effect.

A few approaches to facilitate the detection of salient objects have leveraged object-ness framework [132] without the need of category information. Jia and Han [133] computed the saliency of each region by taking into consideration the overall agreement between salient regions over the whole image, with an explicit emphasis on nodes that are more likely to be foregrounds according to the object-ness criterion. They estimated their regional saliency, called diverse density, as:

$$DD(r_i) = \sum_{j=1}^{N} D_r(r_i, r_j) o(r_j) + (1 - D_r(r_i, r_j))(1 - o(r_j)) \tag{2.20}$$

where $o(r_i)$ is the object-ness score of region r_i computed as spatially weighted average object-ness score of all the enclosed pixels.

Another set of methods in this category consider boundary connectivity in selecting the salient regions [134]. Intuitively, salient objects are much less connected to the image border than the ones in the background. Zhu et al. [134], computed the boundary connectivity score of a region according to the ratio between its length along the image border and the spanning area of this region.

2.3.2.2 Top-down Approaches

These models adopt extrinsic cues to assist the detection of salient objects. In addition to the visual cues observed from the single input image, extra information can be derived from other sources such as ground truth annotations of the training images, similar images, a set of input images containing the common salient objects, depth maps, etc.

The supervised approaches attempt to map the feature vector extracted from each element (e.g., a pixel or a region) to a saliency score $s \in \mathbb{R}^+$ using a learned function $f : \mathbb{R}^D \rightarrow \mathbb{R}^+$, where D is the feature dimension. Assuming a linear mapping function $s = w^T f$, Liu et al. [122] proposed to learn the weights w with the Conditional Random Field (CRF) model trained on the rectangular annotations of the salient objects. They employed a set of features including the local multi-scale contrast, regional center-surround histogram distance, and

global color spatial distribution. Marchesotti et al. [135] described each image I_k similar to the input image I by a pair of feature descriptors $(f^+_{I_k}, f^-_{I_k})$ representing annotated salient and non-salient regions, respectively. To compute the saliency map, the input image is represented as a set of patches $\{p_x\}^P_{x=1}$, each of which described by a fisher vector f_x. The saliency score for a neighborhood $\mathcal{N}_x$ of p_x is estimated as:

$$s(\mathcal{N}_x) = \|f_{\mathcal{N}_x} - f_{BG}\|_1 - \|f_{\mathcal{N}_x} - f_{FG}\|_1 \tag{2.21}$$

where $f_{\mathcal{N}_x} = \sum_{p_x \in \mathcal{N}_x} f_x$, $f_{FG} = \sum_{k=1}^{K} f^+_{I_k}$ and $f_{BG} = \sum_{k=1}^{K} f^-_{I_k}$. Finally, the saliency map for pixel x in $\mathcal{N}_x$ is estimated using:

$$s(x) = \sum_{\mathcal{N}_x} w\mathcal{N}_x . s(\mathcal{N}_x) \tag{2.22}$$

where $w\mathcal{N}_x$ is a normalized Gaussian weight measuring the spatial distance of x to the center of region $\mathcal{N}_x$.

Instead of concentrating on computing saliency on a single image, some algorithms focus on discovering the common salient objects shared by multiple input images. In [136], Chang et al. defined a co-saliency score for each pixel as the multiplication of its traditional saliency score and its repeatedness likelihood over the input images. The repeatedness property was assessed across multiple images that contained some objects in common.

2.3.2.3 Eye Fixation Prediction Models

Salient object detection models aim to detect and segment the most salient object(s) as a whole by drawing pixel-accurate silhouettes, while fixation prediction models are employed to understand human visual attention and eye movement prediction during free-viewing of static natural scenes or dynamic scenes/vidoes [137]. These methods mostly employ motion, optical flow, or spatiotemporal interest points learned from image regions at fixated locations [137, 138].

It is believed that at early stages of free-viewing (first few hundred milliseconds), mainly image-based distinct features direct eye movement and later on, high-level factors (e.g., actions and events in scenes) [106]. These high-level factors, however, may not necessarily be

the same as bottom-up saliency. A human's head in an image, for instance, may not stand out from the rest of the scene but may attract attention. Therefore, some recent approaches have combined these high-level concepts and low-level attributes to scale up current models and reach the human performance.

2.4 Deep Learning

Recent advances in Deep learning [139] have tremendously affected various image-related learning tasks. Deep learning models are based on the idea that representations of observed data are the results of hierarchical abstraction at many different levels [49]. As level moves further up, more abstract information is generated by building on lower level features. The success of these algorithms, in essence, is mainly due to this powerful feature representations which is based on their excellent capability to discover sophisticated structures in high-dimensional data.

Among the most significant highlights of the improvements in image classification due to advances in large neural networks, are achievements such as the large scale image classification record with the ImageNet database [49] and the DeepFace face recognition method by Facebook [140].

2.4.1 Convolutional Neural Networks (CNN)

Billions of parameters are involved in the deep networks, which requires a large scale dataset for training. To this end, convolutional structure is usually adopted to reduce the number of parameters, such as CNN, whereby small portions of an image share the weights with respect to the spatial relation between adjacent pixels. It is an end-to-end system, in which the input is a raw image, while the output is a prediction through the distinctive features extracted via intermediate layers. With different designs of network architecture, CNN is able to learn various presentation of original input image.

Compared with many traditional methods which depend on hand-engineered features or separately trained by machine learning algorithms, CNN exhibits its powerful ability of

extracting features automatically and optimizing the whole system conveniently.

Many advanced techniques have been coupled into the CNN structure, such as dropout, maxout and max-pooling. The architecture of CNNs usually is a stack of convolutional, non-linear, pooling and fully-connected layers, followed by a loss function layer. This architecture is displayed in Figure 2.3 and outlined in the following subsections. A CNN's learnable parameters are located in the convolutional and fully-connected layers and are tuned through gradient descent. The pooling and Relu layers do not have any parameters because they perform fixed functions.

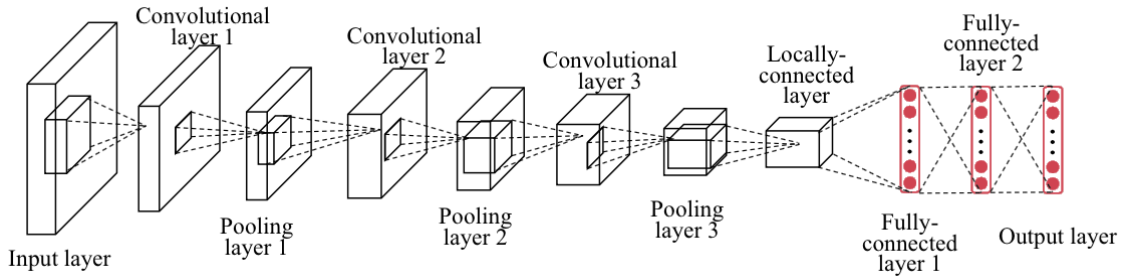


Figure 2.3: Basic structure of CNN

2.4.1.1 Convolutional Layers

Convolutional layer is a core building block of CNN, which differs CNN from traditional artificial neural networks, by defining a kernel to filter the input data. They are used to convolve the input data with multiple filter masks (kernels) to extract features and feed the activation function to generate the output feature maps, which are also the input of the next layer. There are two important concepts: local connectivity and parameter sharing.

Local connectivity means that each filter will convolve only a local region of the input volume in order to decrease the number of the weight parameters. It is inspired by biological systems. The spatial extent of local connectivity is a hyper-parameter called the receptive field.

Parameter sharing means that weights connecting neurons to receptive fields are the same, namely, using the same filter to convolve the entire feature map. One reasonable assumption

is that if one feature patch is useful for a region of an image, it will have the same impact on another region. So parameter sharing not only dramatically reduces the number of parameters but also leads to translation invariance which capture statistics in local patches, e.g., similar edges may appear at different locations.

Let ${x_i}^l$ denote the i^{th} input feature map of l layer, ${k_{ij}}^l$ the kernel connecting the j^{th} feature map of the output layer to i^{th} feature map of the input layer and ${b^l}_j$ an additive bias. Hence, we have

$$ {x_j}^l = f(\sum_i {x_i}^{l-1} * {k_{ij}}^l + {b^l}_j) \tag{2.23} $$

where f is an activation function, usually a rectified linear function.

Through layer by layer convolutional operation, we are able to learn progressive levels of features. For example, the first layer is low-level features, such as edges, lines and corners.

2.4.1.2 Pooling Layers

In order to increase the robustness of CNN to handle translations, pooling layers are usually used after the convolutional layers. They spatially combine convolution layer's outputs, which is related to classical spatial pyramid matching [141]. Its function is to dramatically reduce the spatial size of the feature maps as well as the amount of the parameters leading to efficiently computation in the network, and therefore also control overfitting. The max value or the average value of a local feature map is widely used as a pooling technique. It is formally denoted as

$$ {x_j}^l = down({x_j}^{l-1}) \tag{2.24} $$

where *down(.)* is a sub-sampling function.

2.4.1.3 Relu Layers

The Relu layer is used to gain the non-linearity of the network. This layer uses the non-saturating function as $f(x) = max(0, x)$, which has the non-linear property and has no affection of the receptive fields of the convolution layer.

2.4.1.4 Normalization Layers

Normalization layers give better generalization. Denoting by a_i the single value of i^{th} feature map, the response-normalized activity b_i is given by

$$b_i = \frac{a_i}{(k + a \sum_{j=max(0,1-n/2)}^{min(N-1,i+n/2)} {a^2}_j)^\beta} \tag{2.25}$$

where the constants k, n, a, and β are hyper-parameters.

2.4.1.5 Fully Connected Layers

Several fully connected layers at the end act as nested linear classifiers. The difference between a fully connected layer and a convolution layer is that the perceptrons in the convolution layer are connected only to a local region in the input, whereas all the perceptrons in the fully connected layer are connected to all the perceptrons of the input (input to the fully connected layer).

The classifier most used in the final layer is softmax. The input of the softmax classifier is the feature vector with fixed dimensions, which is the output of the previous layers, and the output of softmax are the probabilities of the mutually exclusive categories, in which the highest one is corresponding to the predicted category.

2.4.1.6 CNN Loss and Optimization

A CNN has the same options as a traditional neural network when selecting a loss function; SVM and cross-entropy loss functions are comparable choices. CNN represents a single differentiable function for which gradient descent is performed during optimization. A key difference is that nodes in a depth slice of a convolutional layer, share parameters. As such, they compute and combine their individual gradients to update the shared parameters. In addition, CNNs that contain pooling layers must keep track of which pixels they downsample in order to backpropagate the gradient.

2.4.2 CNN Architectures

By going deeper with the convolutional networks, CNN dominates the performance in various applications such as image classification and pattern recognition. Architerctures like AlexNet [49], Overfeat [142], GoogLeNet [143], and ResNet [144] attempt to press the limits of what can be fit into a GPU memory and acceptable training time to maximize classification accuracy.

AlexNet is a large, deep convolutional neural network proposed by Krizhevsky et al. [49] which won the 2012 ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) with top 5 test error rate of 15.4%. This was the first time a model performed so well on a historically difficult ImageNet dataset. The network was made up of 5 convolutional layers, max-pooling layers, dropout layers, and 3 fully connected layers, and was used for classification with 1000 possible categories.

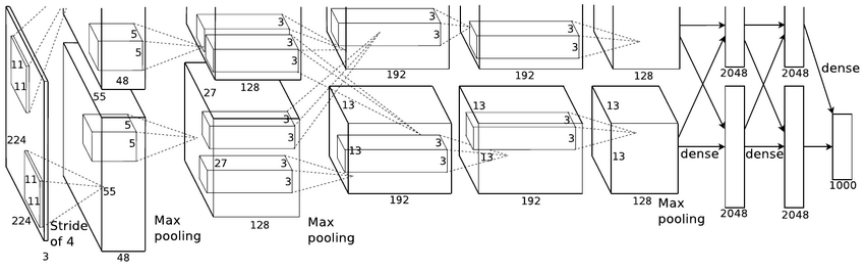


Figure 2.4: AlexNet architecture [49]

Overfeat model [142] shows that by training a network to perform classification, detection and localization simultaneously, the accuracy of each task will be improved. Also it proposed a novel method to localize and detect by accumulating predicted bounding boxes.

The classification architecture used for OverFeat network is similar to that of AlexNet [49], maintaining an 8 layer design of which 5 layers are convolutional and 3 layers are fully connected. The first two convolutional layers and the last convolutional layer are a combination of convolution and max pooling layer, the other two convolutional layers are

without sub-sampling steps. Overlapping of sub-sampling steps are abandoned for faster implementation and the fact that it does not increase classification result by a great lot. Larger feature maps are, also, built in the first two convolutional layers with a smaller stride, this is a trade off between accuracy over training time.

GoogLeNet In order for a convolutional neural network to have better recognition and classification result, a most straightforward way is to enlarge the network. Here enlarging the network refers to increasing the depth of the network by integrating more layers of convolution and also the width by having more units within each convolutional layer. But larger size of a network means much more internal parameters to train which can lead into overfitting and higher computational cost. Faced with this dilemma, Szegedy et al. [143] modified the architecture from fully connected to sparsely connected, both on layer level and inside the convolutional layer level. GoogLeNet is essentially a CNN architecture with Inception modules (Figure 2.5) composed of multiple convolutional filters in 1×1, 3×3 and 5×5 sizes alongside each other. The Inception module provides the option to make the choice of having a pooling or conv operation in a parallel form.

GoogLeNet is a 22-layer network consisting of Inception modules stacked upon each other,

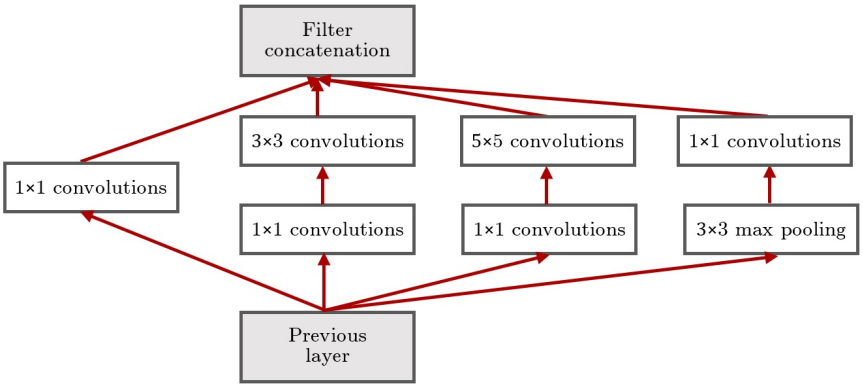


Figure 2.5: Inception module [143]

with occasional max-pooling layers with stride 2 to halve the resolution of grid. This architecture was the winner of ILSVRC 2014 with a top 5 error rate of 6.7%.

ResNet is a network that is up to 152 layers deep, and uses "shortcut connections" between layers in the network to prevent vanishing or exploding gradients, a problem that arises during backpropagation in large-scale networks. He et al. [144] tackled these problems by modifying the architecture leaving the network to learn only a residual mapping (Figure 2.6), and add the input to this output to recover the original mapping. In other words, with this structure there are two possible cases during learning in the optimization of weights: if the identity is the optimal mapping, it is easy to set weights to zero, or if the optimal mapping is closer to identity, it is easier to find small perturbations. This makes a good optimization of weights also with a deeper architecture.

Aside from the new record in terms of number of layers, ResNet won ILSVRC 2015 with an incredible error rate of 3.6%.

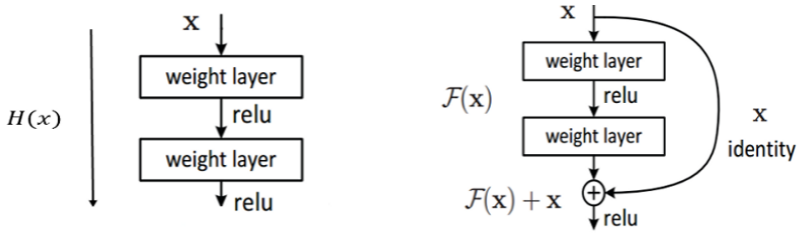


Figure 2.6: Normal CNN vs. CNN with residual learning [144]

2.4.3 Transfer Learning

A deep learning framework usually needs huge amounts of data to train in order for the cost function to converge to a good local minimum point and avoid overfitting on the training set. For some tasks such as fine-grained classification where the size of the dataset is significantly small, a process known as transfer learning [145] can be used as a powerful tool to enable training a large target network without overfitting. Additionally, it can be employed for repurposing learned classifiers for new tasks [146]. A typical way to perform transfer learning or domain adaptation is to train a network from a general large-scale dataset, in which it is believed that features learned are fairly general, and then fine-tuning the network parameters on the target dataset, by retraining a subset of the base

network's learned weights and features. The overall effect is a classifier that fits the new dataset with significantly less work than retraining a new network.

2.5 Vehicle Recognition and Classification

For the vehicle recognition tasks in ITS, there are three important subjects being researched: Vehicle Detection, Vehicle Type Classification, and Vehicle Make and Model Recognition. Although these three categories have different goals, they share certain similarities related to their corresponding level of granularity, and researchers have done numerous works in these areas.

2.5.1 Vehicle Detection

With the development of computer image processing technologies, the problem of detecting vehicles, often from fixed viewpoints, has been well investigated by many researchers. It has become, along with detecting people, bicycles, and other objects, a central component of the PASCAL VOC (Visual Object Category) challenge [53]. The objective of vehicle detection approaches is to find a vehicle ROI over the given image, such that it outlines the vehicle (or vehicle's front/rear face) by filtering out the background regions. This process forms the first step in video-based analysis of different ITS applications and highly affects the accuracy of future tasks mentioned earlier (section 1.1.2). The research efforts in this field can be divided into motion-based and appearance-based techniques.

Motion-based techniques use motion cues to distinguish moving vehicles from static background. Those methods are generally based on frame differencing [147], background subtraction [148] or optical flow [149]. Faro et al. used a background subtraction technique to separate possible vehicle pixels from roads and then applied a segmentation scheme to remove partial and full occlusions among candidate vehicle blobs [150]. Temporal difference is highly adaptive and efficient. However, it cannot cope with noise, rapid illumination variations, or periodic movements in background. Also its performance degrades on slow and fast motion and it cannot extract all the relevant motion pixels. Additionally, the main

drawback of some background subtraction techniques is that they cannot handle complex environments with multiple objects of variable motion. Optical flow-based methods, on the other hand, are less susceptible to occlusion. They can provide accurate sub-pixel motion vectors that are best suited in presence of camera motion, light variation and complex or noisy background. However, the iterative calculations lead to computationally complex methods.

The second approach to vehicle detection uses coded descriptions to characterize the visual appearance of the vehicles. In [151], Jia and Zhang used edge features to model roads and vehicles and then detected possible vehicle candidates by maximizing the posterior probability via the Markov Chain Monte Carlo method. Tzomakas and Seelen suggested that the shadows of vehicles are an effective clue for vehicle detection [152]. Teoh and Bräunl [153], used a multi-sized symmetry searching window to extract each symmetrical region, which was then verified to be a vehicle by an Adaboost classifier. In [154], Wu et al. used wavelet to extract texture features to localize candidate vehicles. These candidate vehicles were verified by a principle component analysis (PCA) classifier. In [155], Ozuysal et al. employed an initial bounding box of a car to select a view-specific classifier to refine the hypothesis. Haar-like and motion features have also been used to detect vehicles in highway [156], and in urban environment [157].

Some approaches combine object detection with pose estimation; they aim to get car configurations with detailed output to describe a more meaningful car shape rather than a bounding box [21,158]. Li et al. [159] proposed a system to learn the appearance of features located at key points (wheels, corners) of a car. This system used an elastic shape model to detect both the location and pose of cars in street images. Sivaraman et al. introduced the idea of vehicle detection by independent parts (VDIP) in [160] for urban driver assistance. In [161], Buch et al. proposed a 3D model-based method for vehicle detection.

2.5.2 Vehicle Type Recognition

Vehicle Type Classification focuses on recognizing vehicles in categories such as Bus, Micro-bus, Mini-van, Sedan, SUV, Bicycle, and Truck. This application finds its utilization in traffic monitoring, restrictions of access, etc. Image-based approaches addressing this task roughly fall into two categories. The first category of methods extract appearance features [162, 163], whereas in the latter, they compute the vehicle's 3D parameters such as size, silhouette dimension and aspect ratio, to recover the 3D model of the vehicle [164, 165].

Oriented-contour point model was proposed in [166] to represent vehicle type. The authors used the edges in the four vehicle orientations from the front view together with a voting algorithm and Euclidean edge distance and achieved classification rate of 93.1%. Thakoor et al. [167] used a Structural Signatures feature that captures the relative orientation of vehicle surfaces and the road surface to classify passenger vehicles into sedans, pickups, and minivans/sport utility vehicles in highway videos with 90% accuracy. Ma and Grimson [168] introduced a rich representation for vehicle classes based on edge points and modified SIFT descriptors to classify vehicles into two classes, i.e., cars vs. mini-vans and sedans vs. taxis using a constellation model. This broad classification is useful for counting but very limited for high-security applications.

Kafai et al. [169] presented a Bayesian network for the vehicle type classification, with features extracted from images of the rear of the car. The resulting feature vector consists of a collection of geometric parameters of the vehicle; including simple features such as the vehicle width, height and more complex features, such as the distance from the license plate to the tail lights. The authors report 10-fold cross-validated classification accuracy of 95.7% for a database of 177 vehicles from four categories of Bus, Truck, Van and Small car. In [170], Matos et al. used a neural network with a collection of image-based features, such as width, height, perimeter and fractal dimension, as inputs. The actual classification was done in two stages using two neural networks, which could achieve an accuracy of 69% on a database with 100 vehicles. Leotta et al. [171] proposed a deformable vehicle model with more details, represented by a polygon mesh with a body and four wheels. They were

able to detect vehicle parts, including the windows, headlights, taillights, grilles, hubcaps, and license plates, using this deformable model. Then the vehicles were classified based on the recovered shape information. Chen et al. [172] proposed to use SVM and random forests to classify vehicles on the road into four classes, namely, sedan, van, bus, and bicycle/motorcycle.

Generally, the advantage of model-based approaches is a reduction in view-point dependency; however, they remain limited to the basic classes and producing simple 3D models accurate enough to distinguish between the many makes and models or subtypes of different models seems unlikely to have high success. Additionally, in vehicle type classification applications, the inter-classes difference is quite large, while regarding the car make and model recognition, the appearances of various models are very similar.

2.5.3 Vehicle Make and Model Recognition

The task of Vehicle Make and Model Recognition (VMMR) is the most advanced use case of cars understanding, with high sensitivity on details, environment changes, rapid variations in manufacturer production and maintenance. However, the amount of relevant scientific literature is relatively small.

The general technique for VMMR contains two parts: feature extraction and classification. The existing approaches cover traditional classification methods (e.g. discriminant analysis [173], Bayesian methods [174], and Support Vector Machines [175]), as well as tools from the computational intelligence area (e.g. neural networks [176], and systems based on various combinations of neural networks, fuzzy sets and genetic algorithms [177]).

In Table 2.1 we have provided a summary of some existing works with details on vehicle view, features, classifier ensemble, number of samples, number of classes and recognition performance. Generally, the literatures fall into three major classes: appearance-based, feature-based, and model-based methods.

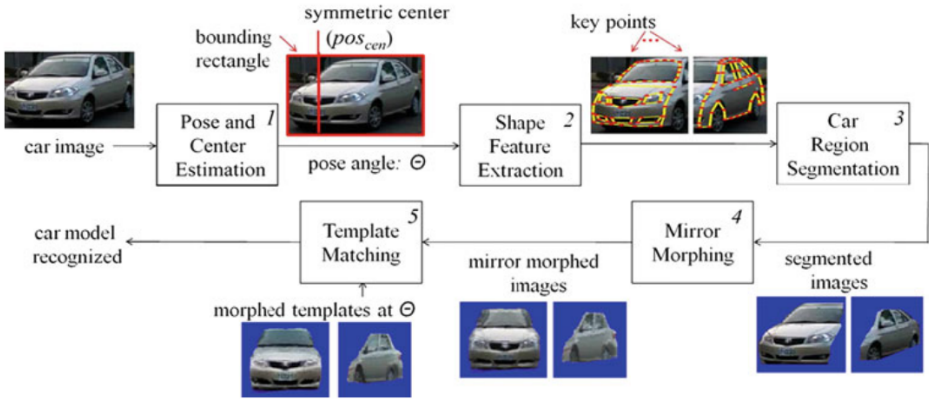
Appearance-based Approaches identify cars by their inherent features including dimensions, shapes, and textures. These methods rely on the pose and position of the

cameras.

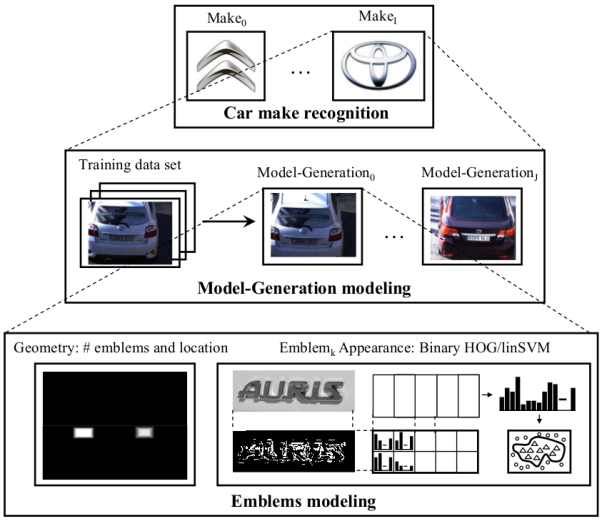
Gu et al. claimed that the recognition of car models could be improved based on shape information and used it to recover the vehicle pose [178]. They evaluated their method on a dataset of 6000 images for 2 target models (Figure 2.7(a)). In [179], Betke et al. utilized symmetry properties and rear lights to detect and track multiple vehicles using a moving camera. A shape based approach was presented in [180], where cars were identified using features extracted from the car back lights and measurements from the car global shape. Llorca et al. [181] proposed the use of geometry and the rear-view image of a vehicle for this purpose. Their approach achieved 93.7% accuracy on 1342 images of 52 makes and models (Figure 2.7(b)).

Feature-based Approaches classify car models using local or global invariant features, and hence their performance depends on the reliability of these features. In such approaches low-level features such as edge-based features [182, 183], contour point features [166], contourlet transform features [176] and corner features are used in the process as well as high-level features, such as, Scale Invariant Feature Transform (SIFT) [184], Speeded Up Robust Features (SURF) [185, 186], Histogram of Gradient (HoG) [187], Pyramid Histogram of Gradient (PHOG), and Gabor features.

In [183], Munroe and Madden used edge features with K-means to distinguish between 5 vehicle classes with 97% accuracy. The algorithm proposed in [188], works by creating an edge map for images present in the reference database. The authors incorporated the appearance and local shape descriptors, calculated for local edge points, to recognize 10 car models, and were able to obtain 70% correct recognition rate. In the method proposed in [189], cars make and model were identified from frontal images. Different types of features were extracted from ROIs extracted relative to the license plate location, and compared to cars the database. Experimentally, square mapped gradients, which are gradients formed from vertical and horizontal sobel edge responses, showed to have a better performance compared to other features on a simple kNN classifier. Their selected discriminative mechanism provided a 93% recognition rate using 1132 images from 77 distinct models. Pearce and



(a) Shape features estimation through Active Shape Model [178]



(b) Incorporation of geometry and the rear-view appearance features [181]

Figure 2.7: System layouts of some appearance-based approaches addressing the vehicle make and model recognition task

Pears [182] examined and compared feature extraction approaches of Canny Edge, Square mapped gradients and Harris corners, and proposed an improved version of Harris corners named as Locally Normalized Harris Strengths (LNHS). In their proposed method, an image is recursively divided into quadrants in which all edges are hierarchically summed and normalized, which achieved a correct classification rate of 96% with the Naïve Bayes classifier, on a dataset of 74 different makes and models. Clady et al. [176] presented a framework for multi-class vehicle type identification based on oriented contour points. In their framework, three voting algorithms and a distance error allowed to measure the similarity between an

input instance and the database classes, which can be combined to design a discriminant function. They attempted to cope with partially corrupted data (car images containing noise caused by barrier), collected in uncontrolled environment, containing 830 images of 50 classes of frontal car images. Their method was able to identify these vehicles with 93.1% accuracy. Negri at al. [166] proposed feature arrays containing contour information, as maps modelling different classes of cars in their MMR scheme. These oriented-contour points matrices, were obtained as a results of a complex process which starts from Sobel filtering and ends with special iterative voting procedure to find such contour points which are invariable for all the training images of a given class. Kazemi et al. investigated three different transform domain feature extractors in their MMR scheme [190], and showed that the best recognition rate on a standard kNN algorithm can be obtained using Curvelet Transform [191].

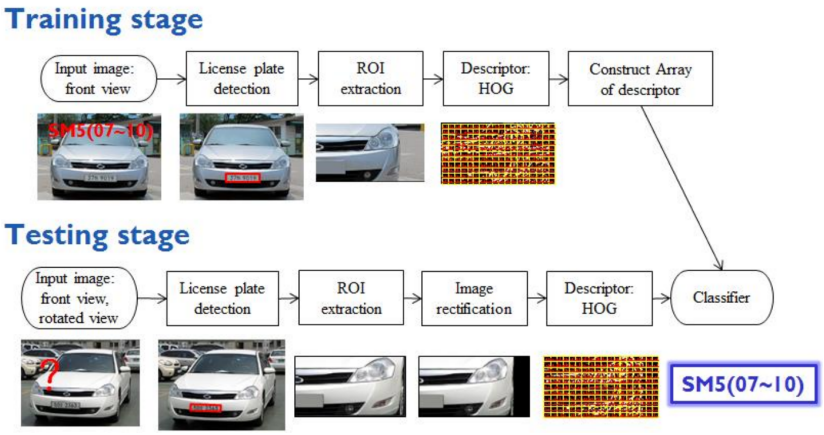
In [187], a ROI was first extracted based on the license plate using a character detection algorithm connected with outer extraction of outer rectangle and symmetry. HOG was used as feature descriptor with kNN and SVM tested as classifiers. This method achieved 100% of positive recognition rate on a dataset of 40 classes and 400 images (Figure 2.8(a)). Baran et al. [186] used local features like SURF to build a dictionary, which was then used to represent vehicle images as sparse vectors of occurrence counts of the words of a dictionary. They utilized a simple multi-class SVM trained over the sparse occurrence vectors. On a dataset containing images of cars front views grouped into 17 classes (one for each car model), their method could get a correct recognition rate of 97.2%. Jang et al. [192], also, attempted to tackle the problem by combining SURF features and bag-of-words model [56] to efficiently search a database of toy car imagery containing 20 car types with 8 side-view images in each category. The matches, were then re-ranked with a structural verification technique. In [193], Fraz et al. formed a lexicon comprised of all training image's features as words. These words were Fisher Encoded Mid-Level-Representation (MLR) of image features such as SIFT, classified using and Euclidean distance based similarity (Figure 2.8(b)). Zhang [194] studied two feature extraction methods for description of frontal

images of vehicles, and found that the Pyramid Histogram of Oriented Gradients (PHOG) has the superiority in its description of more discriminating information, on a dataset of 600 images of 21 vehicle brands, achieving 98.6% classification accuracy. Sarfraz et al. [195] proposed the use of Local Energy based Shape Histogram (LESH) features to represent the class of vehicle. LESH features were computed on ROIs taken from the front view of the vehicles. These features were modeled in a similarity feature space using a probabilistic Bayesian framework. Using Bayes rule, the posterior over possible matches was computed and the highest score was selected as the top matched make and model class. A similar technique was presented in [196], where a local LESH description was extracted for only salient regions.

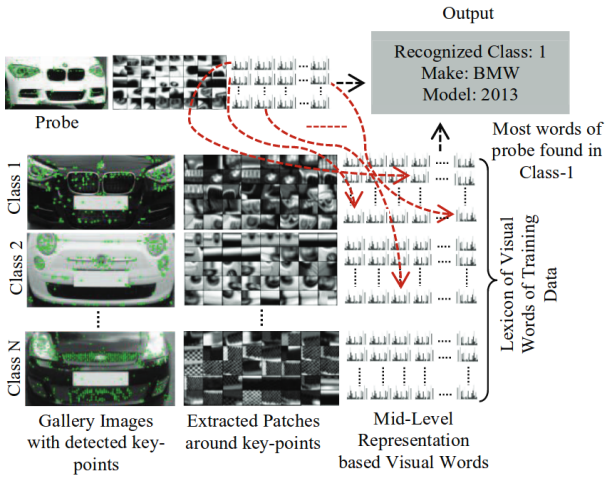
In some techniques, both low-level and high-level features are combined, such as integration of wavelet and contourlet features [197] or PHOG and Gabor features [194].

Model-based Approaches Recently, model-based methods have achieved high precision in vehicle identification. This category of approaches take a different route, and follow the intuition that distinctive features of a fine-grained category, such as the characteristic of the grille of car, are most naturally represented in 3D object space, comprising both the appearance of the features and their location with respect to an object.

A view-independent model-based, top-down approach, proposed by Prokaj and Medioni [198], focuses on pose estimation. The authors suggested to use 3D models of vehicles, fit them to the recognized pose, project them to 2D and use SIFT-like features for the comparison of the vehicles. Their method showed to be 90% accurate on a dataset of 36 classes. Krause et al. [199] improved two state of the art 2D object representations, spatial pyramid matching [141] and BubbleBank [58], to be applicable to 3D with respect to both the appearance and location of local features (Figure 2.9(a)). Additionally, instead of employing annotated training data, they leverage existing 3D CAD models for the basic-level object class of interest, without the need for any manual intervention. They performed their experiments on 196 car models achieving 94.5% accuracy. Hsiao et al. [16] represented each car model by a set of non-parametric 3D curves. They constructed these space curves



(a) LP detection and image rectification using symmetry features [187]

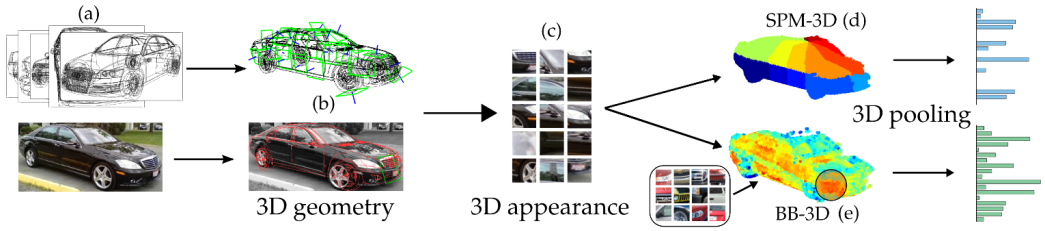


(b) Mid-level representation-based lexicon [193]

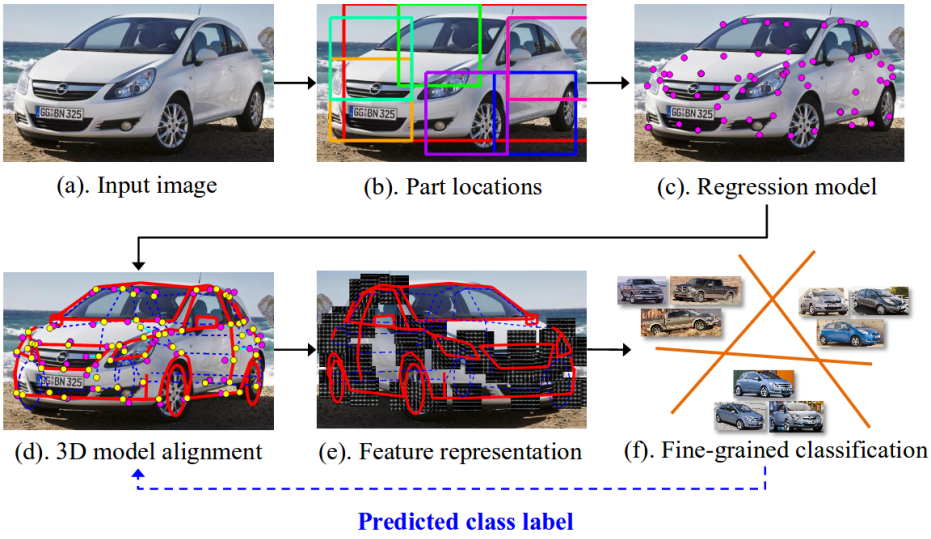
Figure 2.8: System layouts of some feature-based approaches addressing the vehicle make and model recognition task

using 2D training images, and then matched the 3D curves to 2D image curves using a 3D view-based alignment technique. In order to align the 3D curve model to a new test image, they found the closest training image and then aligned the corresponding partial 3D curve model to the edges in the test image by estimating a rigid 3D perspective image transformation. This approach makes their system capable of verifying a type from an arbitrary view. They performed their experiments on a datset of 8 classes with 87% Correct Recognition Rate. In [200], Lin et al. optimized 3D model fitting and fine-grained classification jointly, to obtain positions of landmarks and achieve much better results than other methods on

their own dataset, FG3DCar, consisting of 300 images with 30 disserent car models. they could obtain 90% classification accuracy in their selected setting (Figure 2.9(b)).



(a) 3D geometry along with 3D SPM and BB features [199]



(b) Simultaneous optimization of 3D model fitting and fine-grained classification [200]

Figure 2.9: System layouts of some model-based approaches addressing the vehicle make and model recognition task

Many of the above methods have good performance when dealing with the classification of a few number of vehicle makes or models, resulting in a tight limitation of application [181]. But when it comes to a large number of vehicle makes, their performance usually cannot meet the requirements in realistic applications. Moreover, most of these works heavily rely on hand-crafted low-level features which might not be saliently distinctive among different subordinate-level categories that have extremely similar appearance.

Deep learning-based Approaches To address the fine-grained recognition problem more specifically, recently deep networks are being used to extract discriminative hier-

archical features from large-scale datasets for VMMR [201].

Sochor et al. [202] trained a CNN with 3D vehicle bounding box, its rasterized low-resolution shape, and information about the 3D vehicle orientation besides the vehicle image, to boost the recognition performance on low-resolution surveillance images (Figure 2.10). Their experiments on a dataset of 63750 surveillance images of 27 different makes showed that adding such information achieves an average precision of 80%. Gao and Lee [203] proposed an approach based on deep learning of principal components. They transformed the frontal view of a car to its feature mapping using PCA. They, then used a deep network with three layers of restricted Boltzmann machines to recognize the car make and model. Their system obtained 100% accuracy with regard to the detection accuracy on 107 car models with 30 images for each class. In [204], they proposed a framework in which the frontal views of vehicle images are first extracted and fed into a local tiled convolutional neural network (LTCNN) for training and testing. Their proposed architecture provided the translational, rotational, and scale invariance as well as locality. Yu et al. [205] integrated a CNN model and joint bayesian network to tackle the fine-grained vehicle classification problem to do both vehicle detection and meta information extraction. They achieved a classification accuracy of 89% on a dataset of 208 car models with 200,000 images. In [206], Fang et al. presented a coarse to fine CNN for fine-grained vehicle model recognition, in which the most discriminant parts are automatically detected via feature maps generated by the network. They extracted features from both the global and local regions, respectively for implying holistic cues and describing subordinate-level variation. Based upon the learned features, an one-versus-all SVM classifier is applied for classification, achieving 98.29% accuracy over 281 vehicle makes and models (Figure 2.11). Yang et al. presented a large dataset for fine-grained vehicle detection, and then exploited a CNN model to validate the usefulness of the dataset [21]. They demonstrated several applications including car model classification and verification and achieved a Top-1 and Top-5 classification accuracy of 76.7% and 91.7% respectively. They could achieve a good result on fine-grained vehicle detection, using their manually-collected dataset of 163 car models and 136,727 images. However, for

deep learning networks, this number of images is not enough. Furthermore, the images fed into their model contain only single vehicle, while in reality, images usually have cluttered background.

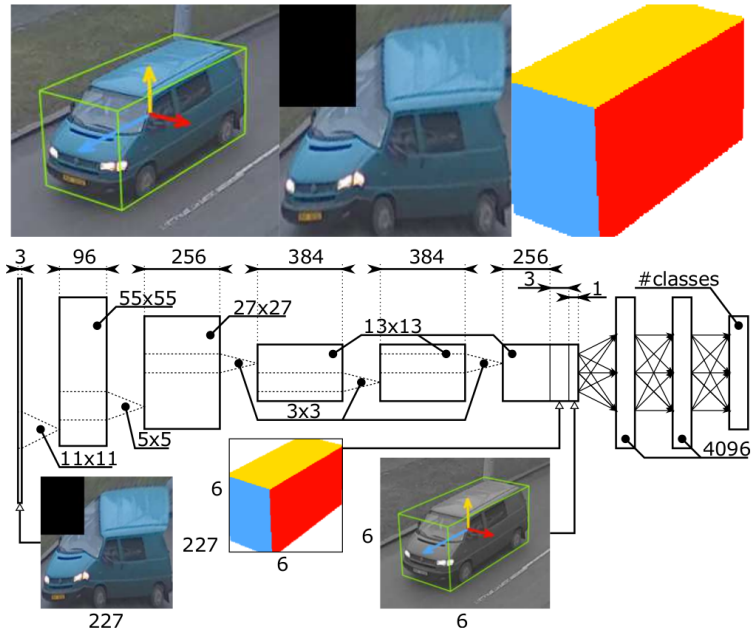


Figure 2.10: BoxCars as input to CNN architecture [202]

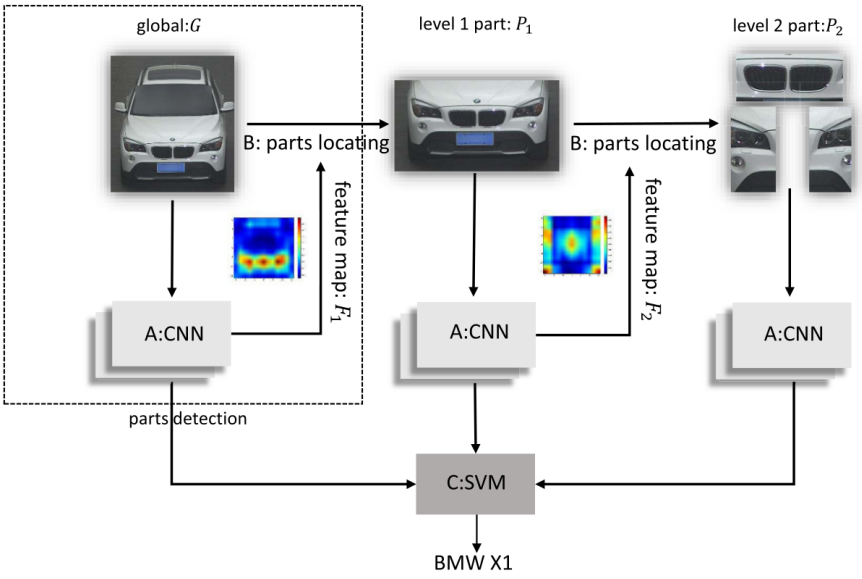


Figure 2.11: CNN-based part detection [206]

Part-based Approaches Recently, some works have focused on the importance of the role of object parts and explored the idea that features extracted from a few important parts may infer more discriminative information than the ones from the whole vehicle image. Liao et al. [207] used Strongly Supervised DPM (SSDPM) to categorize frontal images of vehicles and classification based on discriminative power of different parts of SSDPM (Figure 2.12). They used a dataset of 1482 vehicle images of 8 classes for the verification of their hypothesis, on which they could reach 90% precision. He et al. [208] proposed a recognition framework, which used a part-based detection model to detect frontal view cars and designed an ensemble classifier of neural networks to recognize car models. Their method was tested on a single traffic-camera and could obtain 92.4% classification accuracy on a dataset consisiting of 30 models. In [209], Siddiqui et al. proposed an approach based on the Bag-of-Features paradigm for representing vehicle's front/rear parts.

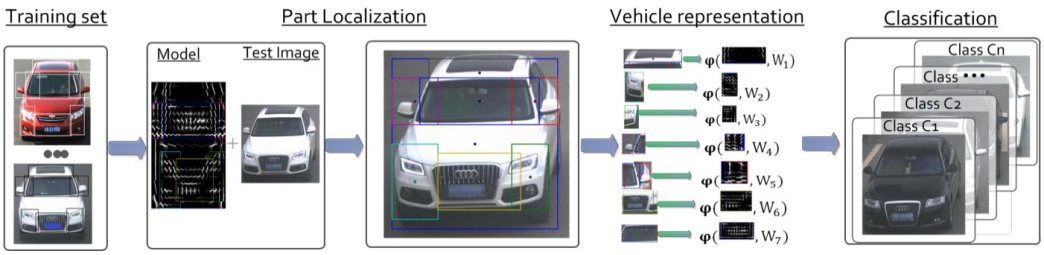


Figure 2.12: DPM-base vehicle part localization [207]

The parts are often localized manually and the part detectors are trained in a supervised manner. To tackle this problem, in [210], Zhang et al. employed an efficient multi-max pooling strategy to generate multi-scale part proposals by using the internal outputs of CNN on object proposals in each image. Then top part proposals were selected at different scales separately, by exploring useful information in part clusters. Finally, the selected part proposals were encoded into a global image representation for fine-grained categorization. This way they automatically detected the key parts of objects in different classes, which have intuitive visualization results and match the rules used by human annotators. They evaluated their method on a dataset of 40 makes and models, and were able to increase the baseline performance of 25.93% for the whole image classification to

40.58% using their part proposals. In [211], Krause et al. proposed to learn discriminative parts of vehicles with CNN and use the parts for fine-grained classification (Figure 2.13). In recognition time, they detected parts and represented their appearances using the learned features, leading to an "Ensemble of Localized Learned Features" (ELLF) representation. This process resulted in 73.9% accuracy on a dataset having 16,185 images of 196 classes of cars.

In another direction, duan et al. [212] presented an approach for make and model recognition that uses MIL to discover local attributes conditioned on viewpoint. To encourage discovered regions to be semantically meaningful, additional constraints on the positions of image regions were applied to the MI-SVM model. To determine viewpoints, an initial set of viewpoints labels was generated with K-means using global image gradient features. These labels were then iteratively updated using a voting scheme which obtains votes from discovered regions in a given image according to the viewpoint of the detector that identified the region.

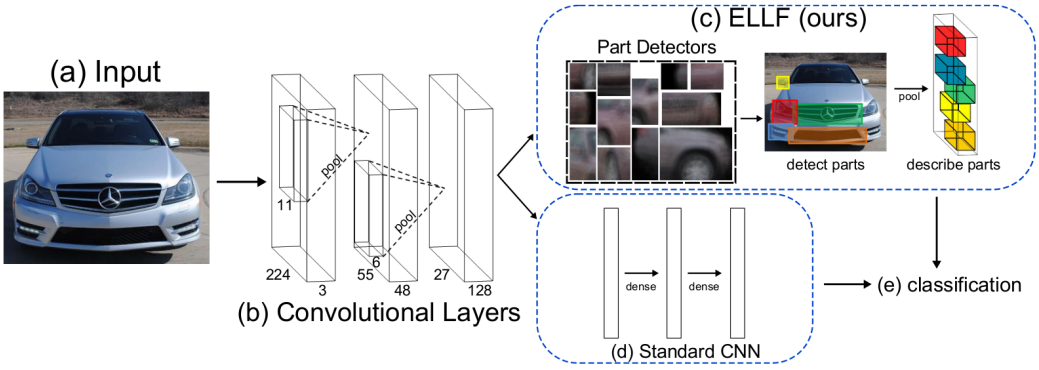


Figure 2.13: Ensemble of Localized Learned Features representation [211]

In the context of vehicle make and model recognition, a few other applications have been addressed in the literature with either to take advantage of various vehicle properties to boost the VMMR system reliability or perform higher level tasks than pure identification.

TABLE 2.1

Summary of previous VMMR works

Ref. Year	Single-Class	Make-Model Viewpoint	Features	Classifier	# Samples	# Classes	Accuracy (%)
2004 [189]	Yes	Front	Square Mapped Graphs	kNN	1132	77	93
2005 [183]	Yes	Front	Canny edges	kNN	180	5	97
2008 [176]	Yes	Front	Oriented contours	kNN	830	50	93
2009 [198]	Yes	3D free	SIFT	Feature matching	400	36	90
2011 [184]	No	Front	SIFT	NN	90	10	54
2011 [182]	Yes	Front	Harris corners	NB	262	74	96
2012 [188]	No	Rear	Shape context descriptors	kNN	400	10	70
2014 [16]	Yes	3D free	3D curves	3D curve matching	190	8	87
2014 [22]	Yes	Front	SURF, HOG	SVM	6936	29	98
2014 [181]	No	Rear	HOG	set of linSVM	1342	52	94

2.5.3.1 Vehicle License Plate Recognition

Automatic License plate recognition systems (ALPR) save time in vehicle-related applications by effectively recognizing the ROI and providing useful information about the vehicle. Since different countries have varied regulations in the format of their license plate, ALPR has been profusely worked on both by individuals as well as institutions around the world. It is generally performed in three major steps: license plate localization, character segmentation and recognition.

Most available algorithms for license plate localization are based either on edge detection or morphological approaches. In [213, 214], edge extraction approaches were employed to detect boundaries of license plate (LP). In [215], a hybrid LP extraction algorithm based on edge statistics and morphology for monitoring highway ticketing systems was proposed.

This approach consists of the following procedures: vertical edge detection, edge statistical analysis, hierarchical-based LP location, and morphology-based LP extraction. In [216], Dubey attempted to improve the results by employing a heuristic approach instead of conventional methods. Local Binary Pattern (LBP) features were employed in [217] to train a boosting classifier for detection of vehicle license plate. An adaptive image segmentation technique using sliding concentric windows (SCW) was considered for LP localization in [218]. The SCW method was developed to describe "local" irregularity in the image, using statistics such as the standard deviation and the mean value as a heuristic for possible plate location. A wavelet transform (WT)-based method was used in [219] for the extraction of important contrast features to be incorporated as guide in searching for LPs. While these algorithms are excellent for the set of images with a knowledge of the LP location, they tend to fail miserably when implemented on other vehicular plates. To implement them successfully on such vehicles, significant alterations are required. In addition, ALPR algorithms should operate fast enough to fulfill the needs of ITS.

2.5.3.2 Vehicle Color Recognition

Color of vehicle is another important attribute that can serve as a useful and reliable cue in a wide range of applications in ITS. However, identifying vehicle color in uncontrolled envirenments is a challenging task due to numerous interference factors, such as haze, snow, rain, illumination or viewpoint variations.

Several methods have employed hand-crafted features such as color sift [220] and feature context [221] to obtain a good performance but are far from producing satisfactory results, particularly in complex real-world scenarios. In [221], Chen et al. proposed a bag of words approach to select region of interest for color recognition. They used feature context (FC) with selected configuration to divide the images into subregions, create histogram for each subregion, and learned it using linear SVM. Some approaches have tackled the problem by classifying vehicle color using 2D histogram features. Baek et al. used 2D histogram of HSV channels to classify colors to 5 classes with SVM [222]. Son et al. [223] proposed

another approach for color recognition using similarity method. They used a convolutional kernel to extract a similarity score estimated based on hue and saturation channels of HSV color space between positive and negative images. These scores were then fed into a SVM classifier. In [224], Hu et al. presented a deep-learning-based algorithm for vehicle color recognition. They combined the CNN architecture with the Spatial Pyramid (SP) strategy to naturally capture the variations in vehicle images while making full use of the structural information of vehicles.

2.5.3.3 Vehicle Logo Recognition

Most work accomplished in the area of vehicle make recognition (VMR) has been through Vehicle Logo Recognition (VLR). This may be viewed as an auxiliary problem to the license plates recognition, as well as to determination of a car type. Research works, mostly, use a knowledge about the location of logo with respect to the license plate and propose various local feature extraction techniques to build discriminative feature representations of logo images. However, it is still a challenge due to difficulties in precisely segmenting the vehicle logo in an image and the requirement for robustness against various imaging situations.

Local Binary Patterns (LBP) [225], Scale-Invariant Feature Transform (SIFT) [45], Edge Histogram descriptors (EHD) [226] and HOG [46] have been studied as features to represent the vehicle logo [184, 227, 228].

Psyllos et al. [229, 230] presented an enhanced SIFT feature-matching technique for vehicle logo recognition. They applied a merged feature matching (MFM) scheme in SIFT to increase feature keypoints. This scheme seems promising but suffers from illumination and viewpoints variations. Yu et al. [231] proposed a system for VLR based on the Bag-of-Words model, which uses a dense SIFT to extract stable features, quantize features by soft assignment, and compute a histogram with spatial information to improve performance. Vehicle logo images are represented as histograms of visual words and then classified by a support vector machine (SVM). Wang et al. [232] detected vehicle logos using edge features,

and then, employed template matching and edge orientation histograms to complete the recognition process. Despite an acceptable reported accuracy, their approach is still limited in realistic environments with shadows and light reflections. Llorca et al. [228] presented a HOG/SVM framework using the gradient distribution as image features. In [233], Liu et al. brought up the Sharpness Histogram Features to calculate sharpness values in the edge map. In [234], ad Spatial Pyramid Matching approach was used to detect logo in natural scenes. Kai et al. [235] proposed hybrid scheme for vehicle-logo localization using appearance features and symmetric property. In [236], Huang et al. introduced a PCA-based pretraining strategy for a CNN logo recognition model.

2.5.3.4 Vehicle re-Identification

Vehicle re-Identification (re-Id) is an important yet frontier topic, which not only faces the challenges of enormous intra-class and subtle inter-class differences of vehicles in multicameras, but also suffers from the complicated environments in urban surveillance scenarios. Unquestionably, license plate is a significant cue for vehicle Re-Id. Nonetheless, it may not work well in unconstrained surveillance scenes due to the various illuminations, viewpoints, and occlusions.

Liu et al. in [237], proposed a Deep Relative Distance Learning (DRDL) model to address the vehicle re-identification problem. The network projects raw vehicle images into an Euclidean space where the L2 distance can be used directly to measure the similarity of arbitrary two vehicles. The input of DRDL are two image sets: one positive set (images of the same vehicle identity) and one negative set (images of other vehicles). In their framework, a coupled cluster loss function pulls the positive images closer and push those negative ones far away. Feris et al. [238] proposed a vehicle detection and attribute-based retrieval system, in which vehicles are searched by attributes like colors and types coarsely. Zapletal and Herout [239] tackled the re-identification problem by using color histograms and histograms of oriented gradients by a linear regressor. They projected and combined the side and front of the 3D bounding boxes of detected vehicles. Then, splitted the combined

image into a grid and calculated the color histogram values for each of the RGB channels and histogram of oriented gradients for each grid cell. In [240], Liu et al. proposed a system for vehicle Re-Id, which fuses the vehicle's color, texture, and high-level semantic features extracted by a fine-tuned deep CNN.

CHAPTER 3

MULTIPLE INSTANCE LEARNING APPROACH TO VMMR

Vehicle Make and Model Recognition (VMMR) plays an important role in Intelligent Transportation Systems (ITS), attempting to incorporate computer vision technologies into vehicles and roadways for monitoring traffic conditions and measurement of traffic parameters such as vehicle count, speed, flow, congestion, etc. We propose two frameworks for VMMR in video sequences. The first one is based on Multiple Instance Learning (MIL) and the second one is based on Convolutional Neural Networks (CNN). The proposed methods can offer valuable situational information for law enforcement units in a variety of civil infrastructures. The proposed systems have the following major components:

Training- This block uses a collection of coarsely labeled images to learn characteristics of each car manufactured per make, model and year.

Testing- This component employs the trained VMMR model to localize and identify different classes of vehicles in query video sequences.

In this framework, a MI classifier is first trained using a comprehensive vehicle make and model dataset. In this process, a set of regions are extracted from each training image as potential discriminant parts of the target vehicle. These regions of interest are image patches with various saliency cues. The ROIs extracted from each class might represent different parts of vehicle such as tail lights, tailgate, bumper, side mirrors, rear window, license plate, etc. Then, some low-level visual features are extracted from the selected image patched. The final feature vectors will be fed into a MIL framework as the instances forming bags. The bags generated from all training images are used to train a MIL classifier.
In a parallel process, an ROI containing the vehicle logo is extracted in each training image.

The process involves a license plate location module, followed by coarse to fine identification of the logo area. Then, another classifier is trained based on the features extracted from these regions.

The resultant model can be used to label the vehicles in each frame of query video with their corresponding make and model by integrating the outputs of logo and MI classifiers. The following steps outline the general process:

1. Identify regions of interest from each image;
2. Extract and fuse features from selected regions;
3. Feed the extracted regions per image as instances of a bag to a MI classifier.

An overview of the system is depicted in Figure 3.1.

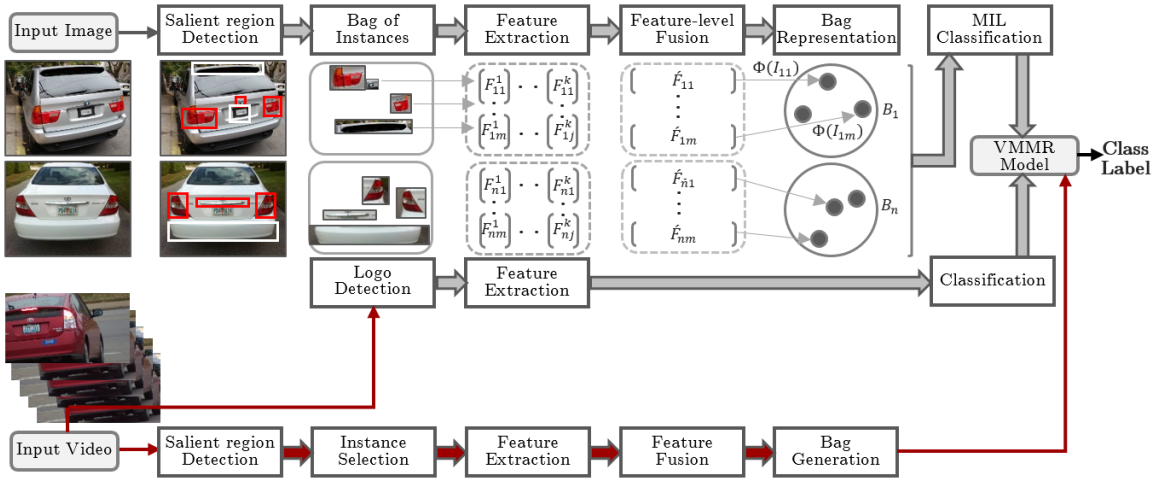


Figure 3.1: Overview of the proposed MIL-based system for VMMR

3.1 Objectives

Compared to generic image classification, fine-grained recognition benefits more from learning critical parts of the objects that can help align objects of the same class and discriminate between neighboring classes. Current state-of-the-art results are, therefore, from models that require part annotations as part of the supervised training process. This poses a problem for scaling up fine-grained recognition to an increasing number of domains,

not to mention it is very unreliable and subjective in a large scale. Vehicles, specifically, own enormous number of designs and model makes that most other categories do not have, enabling a more sophisticated and challenging fine-grained task. In addition, cars yield large appearance differences in their unconstrained poses, which demands viewpoint-aware analysis. Automatically annotating such weakly-labelled training data can be posed as a MIL problem. To achieve a trade-off between efficiency and performance, we need to refine the instances and select a set of instance prototypes from the positive and negative training bags.

3.2 Vehicle Representation

Within MIL framework for VMMR, we can regard each training image as weakly labelled with data such as *Toyota Camry* as a bag containing a set of instances, or possible locations of the vehicle parts. The instances in a particular bag are various sub-images. These sub-images can be segmented regions or image blocks. Objects of interest are considered as positive instances, and the rest are considered as negative instances. If a bag is labeled as *Toyota Camry*, we know that at least one of the sub images contains a salient part of the vehicle. If a bag is labeled as non-*Toyota Camry*, we know that none of the sub images contains a vehicle with that make and model. Each of the instances, or sub images is described as a point in some n-dimensional feature space.

The goal is to find a description which will correctly classify new images as *Toyota Camry* or non-*Toyota Camry*. This can be done by finding what is in common between the positive training images and their differences with the non-positive training images. In other words, from a collection of labeled bags, the discriminative/generative MI learner attempts to induce a concept that will label unseen bags or instances correctly. In addition to learning bag distributions, we expect MIL to infer the label of instance to find distinctive parts of the object of interest (*Toyota Camry*). Figure 3.2 displays a diagram of such MIL framework.

[1]Figure based on [29].

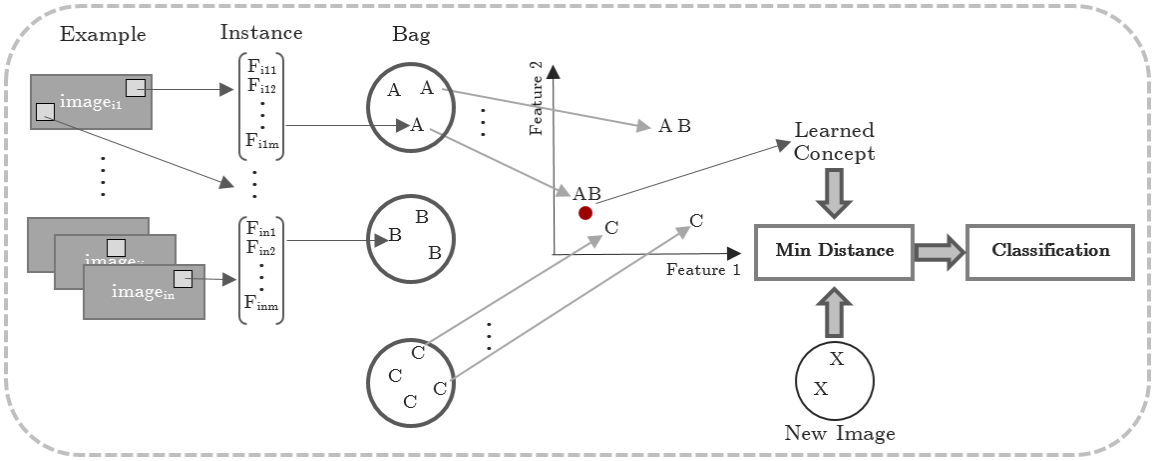


Figure 3.2: MIL system diagram[1]

3.2.1 Instance Selection

Most conventional sub-image retrieval frameworks partition the training and query images into a fixed number of blocks as grids in order to simplify the instance retrieval process and/or take advantage of the spatial information [43, 65]. The positive aspect of these approaches is having control over the number of instances per bag which is fixed for all images. These methods, however, implicitly assume that at least one of the blocks includes the target object. This clearly cannot be the case having objects with various sizes and different resolutions in the dataset. In fact, a rigid partitioning often breaks an object into several blocks or puts different objects into a single block. In case of fine-grained classification, where we need to focus on finer details of the target object, these approaches suffer from noise due to having blocks containing irrelevant background. Also, there is a high chance of having blocks capturing only part of discriminant regions of the object of interest. Considering the limited spectrum of variance between many makes and models, using instances partially representing parts of the target vehicle, would result in a very poor classifier. Figure 3.3(a) displays an example of grid-based sub-image extraction to map an image to a bag with 16 instances. This approach has resulted in splitting parts of vehicle between multiple regions which would override the discriminating power of those regions.

To tackle these issues, more recently, many algorithms [241, 242] have proposed to

partition the image into different regions (that can vary in size and shape) by applying a bottom-up segmentation algorithm [89, 243, 244]. These algorithms aim at capturing local characteristics of each bag by assigning image segments to visually coherent objects and learn the models for the common object and its background. To find the correct correspondence between an image region and the keyword *Toyota Camry*, for instance, a learner must be able to differentiate *Toyota Camry* regions from other noisy regions at the first place. In other words, if the segmentation is ideal, regions will correspond to vehicle parts. But, in general, semantically accurate image segmentation by a computer program is still an ambitious long-term goal for computer vision researchers. In fact, segmentation from low-level cues is often unable to provide semantically correct segments. Such methods may cut the object of interest into multiple components. Therefore, this representation highly depends on the quality of the segmentation because small areas of incorrect segmentation might make the representation very different from that of the real object. A few examples of sub-images generated using different segmentation methods are shown in Figures 3.3(b), 3.3(c) and 3.3(d).

(a) Fixed partitioning (b) K-means [243] (c) Mean shift [244] (d) Gabor Filters [245]

Figure 3.3: Different approaches towards instance representation

The approach we propose does not rely on unreliable segmentation methods for representation. Instead, it captures generic knowledge regarding the typical size and location of distinctive object parts in images, or expresses a relationship between the strength of image edges and the location of object bounding boxes. We use this method to prune the space of discriminant object region locations a priori, allowing us to consider a reduced set of possible regions as positive instances. Our work can also be viewed as a variation of

feature selection methods, in which different features are selected for each example. Given an arbitrary set of unlabeled images, our goal is to discover a relatively small number of discriminative patches at arbitrary resolution which can capture the essence of that data and serve as a fully unsupervised mid-level visual representation. Many semantic entities are just not discriminative enough visually to act as good features. For example in a car, "door" is a well-defined semantic category, but it makes a lousy detector simply because doors are usually plain with similar shape in many makes and models and thus not easily discriminable. The desired patches need to satisfy two requirements. First, they need to be good representatives, meaning that they need to occur frequently enough in the visual world. Second, they need to be discriminative, i.e., they need to be different enough from the rest of the visual world. These patches could correspond to parts, regions, objects, visual phrases, etc. but are not restricted to be any one of them.

3.2.1.1 Salient Region Selection

In this work, we propose to generate parts which can be detected in novel images and learn which parts are useful for recognition. This is done by extracting pertinent, attention grabbing regions of the scene without conscious awareness [246]. We use a model that can handle both eye fixation prediction and saliency detection to identify more discriminative regions instead of the traditional grid or segmentation-based patches. Instead of employing local image properties such as contrast and rarity which have limited ability to model some global perceptual phenomena, it makes use of figure-ground segregation [247].

Based on the Gestalt principles of grouping, humans tend to organize a cluttered image through a process of figure–ground segregation without focal attention, i.e., by identifying those regions of the retinal images that are object-related (figures) for further processing, and assign other regions to the background [248]. It basically relies on the detection of feature discontinuities that signal boundaries between the figures and the background and on a complementary region-filling process that groups together image regions with similar features. Examples of such features include continuity, convexity, symmetry, parallelism and

surroundedness [248, 249]. The method makes use of simple image processing operations to leverage the topological structural information which is scale invariant, and also has a strong influence on visual attention [247].

In this bottom-up Boolean Map-based Saliency model (BMS) [250], based on the fact that an observer's momentary conscious awareness of a scene can be represented by a Boolean Map [251], each image I is first decomposed into a set of Boolean maps $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n)$. These maps are generated by sampling from a prior distribution function $F(\mathcal{B}|I)$ conditioned on the image, over the feature map $\phi(I)$ and the threshold θ:

$$\begin{gathered} \mathcal{B}_i = \textbf{THRESH}(\phi(I), \theta), \\ \phi \sim F_\phi, \theta \sim F_\theta^\phi \end{gathered} \tag{3.1}$$

F_ϕ and F_θ^ϕ denote the prior distribution of feature channel sampling and threshold sampling θ on the feature channel ϕ respectively. The feature channels can represent one or a combination of features like color, intensity, orientation, motion, etc.

For each Boolean map, an attention map $A(\mathcal{B})$ is computed to represent its influence on the visual attention. To compute the map, first an activation map $\mathcal{M}(\mathcal{B})$ is generated by applying binary image processing techniques to activate regions with closed outer contours (unsurrounded regions) on $\mathcal{B}$. Intuitively, a white (black) pixel in $\mathcal{B}$ is surrounded if and only if it is enclosed by the black (white) set. Formally, the surroundedness can be defined based on a pixel's connectivity to the image border pixels.

Then, to further emphasize the regions with rare topographic features, the resultant activation maps are normalized to create the attention map. To do so, each activation map $\mathcal{M}(\mathcal{B})$ is split into two sub-activation maps $\mathcal{M}^+(\mathcal{B})$ and $\mathcal{M}^-(\mathcal{B})$. These sub-activation maps are responsible for activating the surrounded peaks above and below the corresponding threshold. Then, for eye-fixation prediction, an L2-normalization is applied to emphasize attention maps with small active area. Finally, attention maps are averaged into a mean attention map $\bar{A}$ over generated boolean maps:

$$\bar{A} = \int A(\mathcal{B}) \, \mathrm{d}F(\mathcal{B}|I) \tag{3.2}$$

$\bar{A}$ is further post-processed to localize the regions people are likely to look at and output a saliency map S. This process is summarized in Figure 3.4.

The output of both eye-fixation prediction and saliency detection for a few sample images are displayed in Figure 3.5. The red rectangles in third row are the selected image patches after thresholding both maps.

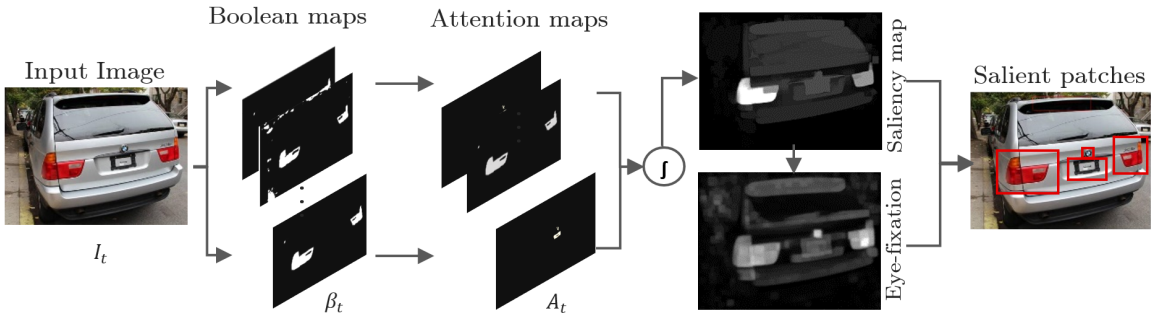


Figure 3.4: Salient region detection

Figure 3.5: Few sample images with their instances selected based on saliency map and eye-fixation prediction

When dealing with noisy images, although salient detection and segmentation methods cannot guarantee robust performance on individual images, their efficiency and sim-

plicity makes it possible to automatically process a large number of images, which can then be further filtered for reliability and accuracy, thus enabling many applications to run robustly [137] and even supports unsupervised learning in the case of our application. Also, by leveraging the surroundedness cue for figure-ground segregation, BMS is less responsive to the edges and cluttered areas in the background. More importantly, with this approach there is no need for spectral transformations, off-line training or multi-scale processing.

3.2.2 Instance Representation

Each image patch is represented by a feature vector and becomes an instance of the bag. Various features have been employed in the literature [46, 252]. The most common features include HOG [46], LLC features [253], Fisher vectors (FV) [47] and deep-learning features [97]. In the current framework, we represent the instances using the FV encoding method which has proven to be very effective in various image recognition tasks [47].

Given a set of T training images, we denote by I_{td} a salient region d extracted from image I_t. For each region, first, local features are extracted using a variation of SIFT descriptors, known as dense SIFT [47]. Standard SIFT descriptors are extracted at local points of interest, usually referred to as key-points. The locations of key-points are determined around corners, areas with high magnitudes, and salient regions in general. If key-points can be extracted correctly from an image, then, a global feature can encode the concept effectively. However, if the key-points do not capture the object of interest within the image, the effectiveness of the SIFT feature can be reduced significantly. To alleviate this weakness, we use SIFT descriptors from a dense multi-scale grid covering the entire extracted patch. Let $DSFIT(I_{td}) = (\mathbf{x^t_{d1}}, \ldots, \mathbf{x^t_{dN}})$ be the set of all extracted dense SIFT features for patch I_{td}, where $\mathbf{x^t_{di}}$ is a standard 128-D SIFT descriptor, d is the index of salient region in image I_t and N is the number of descriptors per image patch. Figure 3.6 illustrates grid partitioning and an example of dense SIFT extraction on a generic image patch.

Given the set of extracted dense SIFT features from all selected patches in training

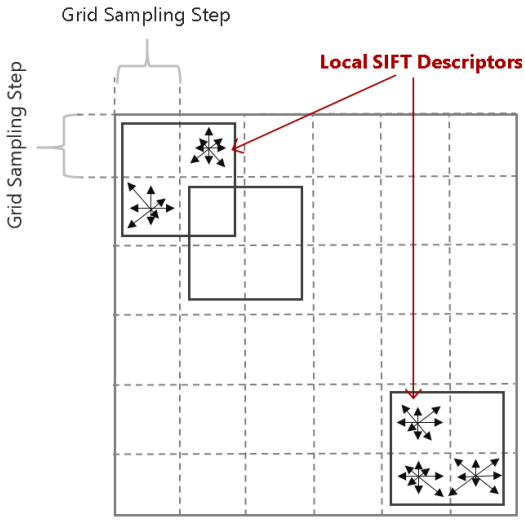


Figure 3.6: An example of Dense SIFT extraction

data, $(\mathbf{x_{di}^t}, t = 1, \ldots, T, d = 1, \ldots, D^t, i = 1, \ldots, N)$, a generative model, such as Gaussian Mixture Model (GMM) with K components, is learned using the EM algorithm [47]. It can be regarded as a *probabilistic visual vocabulary*. Let $\Theta = (\mu_k, \Sigma_k, \pi_k : k = 1, \ldots, K)$ be the parameters of the GMM fitting the distribution of descriptors, where π_k , μ_k, and Σ_k are respectively the mixture weight, mean, and covariance matrix of Gaussian component k. The Fisher Kernel characterizes a local feature of a sub patch by its deviation from the generative model. The deviation is the gradient of the sub patch log-likelihood with respect to the generative model parameters. The GMM associates each feature vector $\mathbf{x_{di}^t}$ to mixture component k with a strength given by the posterior probability:

$$q_{ik} = \frac{\exp\left[-\frac{1}{2}(\mathbf{x_{di}^t} - \mu_k)^T \Sigma_k^{-1} (\mathbf{x_{di}^t} - \mu_k)\right]}{\sum_{t=1}^{K} \exp\left[-\frac{1}{2}(\mathbf{x_{di}^t} - \mu_t)^T \Sigma_k^{-1} (\mathbf{x_{di}^t} - \mu_t)\right]} \tag{3.3}$$

To simplify the notation, let x_{ij} be the j^{th} dimension of $\mathbf{x_{di}^t}$. For each component k, we compute the mean and covariance deviation vectors:

$$u_{jk} = \frac{1}{N\sqrt{\pi_k}} \sum_{i=1}^{N} q_{ik} \frac{x_{ji} - \mu_{jk}}{\sigma_{jk}}, \tag{3.4}$$

$$v_{jk} = \frac{1}{N\sqrt{2\pi_k}} \sum_{i=1}^{N} q_{ik} [(\frac{x_{ji} - \mu_{jk}}{\sigma_{jk}})^2 - 1] \tag{3.5}$$

where $j = 1, 2, \ldots, F$ spans the vector dimensions.

The FV of a given salient region d of an image I_t is the vectorial representation of all

the deviation, estimated through concatenation of $\mathbf{u_k}$ and $\mathbf{v_k}$ for each of the K Gaussian components, i.e.,

$$\Phi(I_{td}) = [u_{11} \cdots u_{F1} \cdots u_{1K} \cdots u_{FK}, \\ v_{11} \cdots v_{F1} \cdots v_{1K} \cdots v_{FK}] \tag{3.6}$$

Thus, each image I_t is mapped to a bag with D^t instances. Each instance is represented by a feature vector $\Phi(I_{td})$ as defined in (3.6). The instance features are FVs extracted from regions detected through a saliency detection model based on figure-ground segregation.

3.3 Vehicle Classification

This data can be fed into any MIL framework. Many MIL approaches were discussed in section 2.2.2. Generative models are based on a strong assumption that all true positive instances form a compact cluster in the feature space. This is, however, not necessarily the case in our application, as the distributions of positive instances can be arbitrary and most likely multi-modal. Hence, learning a single target distribution to represent positive instances is inadequate in capturing their distributions. On the other hand, large margin discriminative methods are much more robust and achieve an improved performance. In this context, with the current assumptions of the method, we can employ various classifiers with no embedded instance selection such as DD-SVM [67], MI-SVM, mi-SVM [68] and Citation-kNN [71].

Once a classifier is trained, we can use it for labeling incoming instance prototypes. One promising aspect of MIL is that it allows for the automatic model learning and instance-level label prediction at the same time. In the end, a discriminative classifier is learned with the simultaneous label predictions on the selected instances. This way we can identify the distinctive regions per image in another level and employ this information later to boost the designed classifier. In mi-SVM [68], for instance, the labels of instances in positive bags are initialized as positive, and then the SVM classifier is trained repeatedly until each positive bag has at least one instance which is classified as positive by the classifier.

3.4 Vehicle Make Recognition Based on Logo

Additional vehicle attributes can contribute to improving the accuracy of VMMR. Vehicle manufacturer logo is a remarkable characteristic of vehicles. The attention mechanism of a human vision system performs in a way that given an image/video containing vehicles, it quickly focuses on certain objects like logos if no prior knowledge of vehicle is available [254]. In fact, logos are widely used visually-salient symbols serving as remarkable identifiers of related organization. The vehicle manufacture's logo is a small object that should be a unique identifier of the car make. It could be found in several parts of the car especially in front and rear areas. Mostly, car logo shape is unique in terms of color, texture, and geometry which make the logo easy to be noted or seen by human. With logos accurately detected, the manufacturer recognition can become straightforward. This can efficiently narrow the search space for VMMR and can boost the reliability of VMMR system.

However, due to the ever-changing road environments, vehicle images are usually captured with great variabilities, caused by different backgrounds and lighting conditions. Motion and varying shooting angles of surveillance camera will cause distortions to the logo shapes. The tiny portion of logo area in a captured image adds extra difficulty to the task.

The majority of the approaches concerning vehicle logo recognition (VLR) make use of license plate (LP) location in either front or rear view images of vehicle, followed by a coarse-to-fine approach to identify the logo ROI. The texture of license plate region is steady-going and fixed in size [255]. If it can be detected quickly and coarsely, it could help in giving a rough position of the vehicle logo in the vertical direction. The layout of logos and other parts of the frontal/rear region of a vehicle are similar in almost all different vehicles. A sliding window technique is then applied in a ROI defined above the detected LP. In this process a classifier function is subsequently applied to the ROIs provided by the sliding window stage, for template matching. Finally, a majority vote approach is implemented to recognize the logo using the binary outputs given by the classifier [228].

3.4.1 Logo Feature Representation

We have employed an enhanced SIFT matching module for detecting and extracting the points of interest in the query logo image [229]. The whole process is fine-tuned by clustering the matched keypoints, using Generalised Hough Transform [256], and geometric verification by means of an affine transformation. In this process we extract SIFT points from all training logo images. For each SIFT feature f_{qi} in query image I_q, the descriptors are used to find the total number of its Nearest-Neighbor (NN) matches among all features in the training database using L-2 Euclidean metric:

$$NN_i = cardinality(arg\{\|D_{qi} - D_{ji}\| < \gamma\}) \tag{3.7}$$

where D_{qi} is the ith descriptor for the query image, and D_{ji} (i.e. $D_{j1}, D_{j2}, \ldots, D_{jN}$) is the ith descriptor for image I_{tj} and $N = 128$. Therefore, the number of NNs in the training database depends on the selection of threshold γ. This parameter is varied to keep the ratio of NNs to the total keypoints fixed:

$$\tau = \frac{NN_i}{KP_q.KP_t} \tag{3.8}$$

where KP_q is the total number of keypoints detected in query image q, and KP_t is the total number of keypoints in the training database.

Not all of the extracted keypoints in each image are important in the matching process. In fact, we have several irrelevant features extracted due to ambient illumination reflections, shadows or noise. Thus, in the next step, the reasonable NN matches are determined to verify whether the query image represents a logo image stored in the database. NN features are clustered using Generalised Hough Transform (GHT) [256], in order to define the parameters for a similarity transformation between the query and the database features. GHT identifies clusters of features with a consistent interpretation by using each feature to vote for all the logo images that are consistent with the feature. Therefore, when the NN distance between a pair of query and database feature is less than a certain threshold, this pair participates in a Hough similarity transformation and a vote is added to the

corresponding cell in the Hough array. The database logo image with the highest number of votes is considered to be the most similar with the query image.

3.4.2 Logo Matching

In the next step, the locations of the keypoints in the query and the database image are verified for geometrical consistency. Those keypoints that fit well to an affine geometrical transformation are called inliers, while those that are inconsistent to it are called outliers. In order to select which candidate points should be used for affine transformation, the RANSAC method [257] is applied to a pair of images by randomly choosing three pairs of matched points of interest. These three pairs of points of interest are fitted to an affine transformation, which is then applied to the rest of the keypoints of the two images. This procedure is repeated for all clusters found in the GHT and the database image with the maximum number of inliers is selected as the matched logo. Figure 3.7 displays sample logos extracted based on the location of license plate matched with a logo in the training dataset. The column on the right illustrates the frequency of images matched from each cluster of GHT.

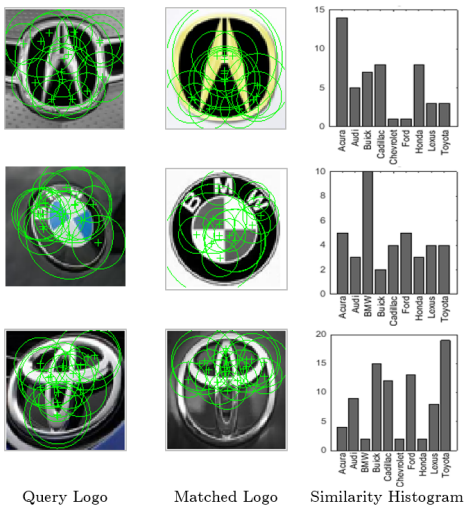


Figure 3.7: The process of logo matching

CHAPTER 4

INCORPORATING MULTIPLE INSTANCE LEARNING INTO DEEP LEARNING FOR VMMR

Feature learning is a promising approach that can generate powerful appearance representations. Much work has focused on encoding low-level features such as SIFT or HOG (section 2.5.3) or mining discriminative templates [4]. The recent success of convolutional neural networks (CNNs) [49] on large-scale classification demonstrates that powerful features can be learned directly from pixels. This inspires us to adopt CNNs for fine-grained recognition of vehicles. In particular, we use a convolutional neural network that accepts pixels as its input and outputs probabilities of classes. The key benefit of employing such deep network is that this enables the classifier to learn higher lever structures within the image. For our particular problem, the network might have the ability to learn the appearance of the headlights on the first layer, their relative positions and distance on the second layer, and their relative location with respect to the car license plate on the third convolutional layer. In other words, the key to success is that the network can learn the feature extraction step in an optimal manner and can avoid the need for manual feature engineering.

4.1 Objectives

The CNNs capability in modeling complex non-linear functions and inferring the context information of neighboring pixels in the input data, has resulted in their state-of-the-art performance on many computer vision tasks focusing on global learning. However, in fine-grained classification, distinctive information comes from local patches which are distributed inconsistently in different classes at various positions and scales. Thus, learn-

ing such combination of discriminative parts could conflict with the CNN traditional global learning scheme and limit its performance in fine-grained classification. Additionally, CNNs require costly supervision; i.e. human annotations is still a key part of many popular frameworks [258]. As mentioned before, fine-grained labeled data is very expensive to acquire, and our goal is to find the subset of regions that are discriminative and use them to train a model. Detailed manual annotations are time-consuming and intrinsically ambiguous. An alternative is to learn local concepts using global annotations, which is the main idea of Multiple Instance Learning (MIL) (refer to chapter 3). In such weakly supervised learning framework, the training set contains labeled bags that are composed of unlabeled instances, and the task is to predict the labels of unseen bags and instances.

In this chapter, we aim to counter the shortcomings of CNN for fine-grained image classification by simultaneously learning the representation to infer the most distinctive instances. We propose a weakly supervised framework for vehicle make and model recognition by combining multiple instance learning loss and deep residual networks.

4.2 Traditional CNN

As a supervised learning method, CNNs take pairs of data-label for training a model. In its supervised learning process, the optimization is done by minimizing the total cost L:

$$L = \sum_{i=1}^{N} L_i = \sum_{i=1}^{N} f_{loss}(\mathbf{y_i}, F(\mathbf{W}, \mathbf{x_i})) \tag{4.1}$$

where $\mathbf{y_i} = \{0,1\}^{1\times C}$ are the training data labels for C classes, $\mathbf{W}$ represents the set of adjustable parameters in the CNN architecture, and $\mathbf{x_i}$ is the input matrix. F indicates a set of functions in CNN, depending on the selected architecture, such as convolutional, pooling and fully-connected functions. The output of $F(\mathbf{W}, \mathbf{x_i})$ is $\mathbf{h^i} \in \mathbb{R}^{1\times C}$ which can be interpreted as the class label of input or probabilities associated with each class.

A very common loss function in convolutional neural networks is softmax, which provides a way of assigning probabilities to each class, with cross-entropy loss function. Let CNN output $\mathbf{h^i} = F(\mathbf{W}, \mathbf{x_i})$, $\mathbf{h^i} = \{h_1^i, h_2^i, \ldots, h_C^i\}$, the predicted label for training input $\mathbf{x_i}$ is

max value of $\mathbf{h^i}$:

$$\hat{y}_i = argmax_{j=1}^{C}(h_j^i) \tag{4.2}$$

with the cross-entropy to measure the prediction loss of the network for softmax activations of p_j:

$$f_{loss} = -\sum_{j=1}^{C} y_j log(p_j), \tag{4.3}$$

$$p_j = \frac{exp(h_j)}{\sum_{l=1}^{C} exp(h_l)}, \quad j = 1, 2, \ldots, C \tag{4.4}$$

The gradient of softmax with cross-entropy loss with respect to the output is

$$\begin{aligned} \frac{\partial f_{loss}}{\partial h_k} &= \sum_{j=1}^{C} \frac{\partial f_{loss}}{\partial p_j} \frac{\partial p_j}{\partial h_k} \\ &= \frac{\partial f_{loss}}{\partial p_k} \frac{\partial p_k}{\partial h_k} + \sum_{j \neq k} \frac{\partial f_{loss}}{\partial p_j} \frac{\partial p_j}{\partial h_k} \end{aligned} \tag{4.5}$$

Computing the partial derivatives yields

$$\frac{\partial f_{loss}}{\partial p_j} = -\frac{y_j}{p_j}, \tag{4.6}$$

$$\frac{\partial p_j}{\partial h_k} = \begin{cases} \frac{exp(h_j)}{\sum_{l=1}^{C} exp(h_l)} - \left(\frac{exp(h_j)}{\sum_{l=1}^{C} exp(h_l)}\right)^2 = p_j(1 - p_j) & j = k \\ -\frac{exp(h_j) exp(h_k)}{\left(\sum_{l=1}^{C} exp(h_l)\right)^2} = -p_j p_k & j \neq k \end{cases} \tag{4.7}$$

Thus, the gradient will be

$$\begin{aligned} \frac{\partial f_{loss}}{\partial h_k} &= -y_k(1 - p_k) + \sum_{j \neq k} y_j p_k \\ &= -y_k + p_k \sum_{j=1}^{C} y_j \end{aligned} \tag{4.8}$$

The gradients for weights is thus

$$\frac{\partial f_{loss}}{\partial \mathbf{W}} = \frac{\partial f_{loss}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{W}} \tag{4.9}$$

Finally, the weight parameters will be updated as

$$\mathbf{W}_{new} = \mathbf{W}_{old} - \lambda \frac{\partial f_{loss}}{\partial \mathbf{W}} \tag{4.10}$$

where λ is the learning rate.

We have used a Residual Network (ResNet) [144] (section 2.4.2), which follows the same procedure in its architecture, but with shortcut connections that skip several layers, to prevent vanishing or exploding gradients. Shortcut connections simply pass through the inputs as outputs and merge them into the ones from the skipped layers. The layers with shortcut connections learn residual and the outputs merged with shortcut connection are

$$\mathbf{y} = F(\{\mathbf{W}_i\}, \mathbf{x}) + \mathbf{x} \tag{4.11}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the layer's input and output vectors, and $F(.)$ represents the residual mapping to be learned.

Our goal is to model deep CNNs in a weakly supervised learning framework. Instead of assigning labels to all generated images, we treat the generated images as a bag and the original label as bag-level label. For binary MIL, a bag is labeled positive if the bag contains at least one positive instance, and it is labeled negative if all its instances are negative.

4.3 Multiple Instance Learning Based CNN

Instead of extracting CNN features, disjointly selecting discriminative parts and then learning a classifier, jointly learning distinctive regions with features would likely bring improvements to both distinctive region discovery and feature learning. MIL would also enable us to further take advantage of instance labeling, since the cost of part detection makes it impractical to train part detectors on a significant number of classes. Therefore we define a new CNN structure called Multiple Instance Learning Convolutional Neural Networks in order to further free the power of deep learning, which is currently constrained by the limited amount of well labeled data. Specifically, we use CNNs to learn appearance descriptors, and perform unsupervised discovery of discriminating regions in a supervised multi-instance fashion.

Instead of feeding the training image into the network and following the global learning approach, we formulate each image into a bag consisting of multiple instances. Let's denote each input as a bag $\mathbf{B}_i$, where within each bag we have a number of instances

$\mathbf{B}_i = \{x_{ik}\}_{k=1}^{m}$. Labels $\mathbf{y_i} = \{0,1\}^{1\times C}$ are only available at the bag level, while labels of instances ($y_{ik} \in \{0,1\}$) are unknown. In such formulation, all instances in each bag share the same label as bag.

Instances are local image patches selected at various locations and scales, and form the inputs to the deep network. For each training image $\mathbf{B}_i$ only the local image patch that has the highest response contributes to the loss function and drives the update of network coefficients $\mathbf{W}$ during the backward propagation. Accordingly, the learned CNN is expected to have high responses on discriminative local patches. In other words, the most discriminative local patches for each image class are automatically discovered after the CNN training.

We reformulate the CNN loss function (4.3) for MIL as

$$f_{loss} = -\sum_{j=1}^{C} y_j log(p(c_j|B)) \tag{4.12}$$

where $p(c_j|B)$ represents the probability of bag B being classified as jth class. Following the MIL settings and the noisy-or model (2.5) [64], we have

$$p(c_j|B) = 1 - \prod_{k=1}^{m}(1 - p(c_j|x_k)) \tag{4.13}$$

where $p(c_j|x_k)$ is the probability that the kth region is classified as the jth class, and based on (4.4) equals

$$p(c_j|x_k) = \frac{exp(h_{kj})}{\sum_{l=1}^{C} exp(h_{kl})}, \quad j = 1,2,\ldots,C,\ k = 1,2,\ldots,m \tag{4.14}$$

where h_{kj} is the jth output of the CNN model before loss layer for kth region.

However, there is a drawback in (4.13); when all instance probabilities of a bag (4.14) for a certain class are small, and the number of instances m_i is large, $p(c_j|B)$ will be close to 1. This way, the model distinguishes the bag as positive, but considering the low probabilities of all instances, it should be classified as a negative bag.

In such scenario, reformulating the probability as

$$p(c_j|B) = \max_k p(c_j|x_k) \tag{4.15}$$

would solve the issue and classify the bag as negative when all instance probabilities are close to 0, and positive when at least one instance has a probability close to 1.

However, in case of fine-grained classification, there are usually more than one instance contributing to the bag label. Therefore, we employ the following weighted probability to highlight the importance of the most positive instance while assigning more weights to the cases where multiple positive instances exist.

$$p(c_j|B) = p(c_j|x_{k_j}) \sum_{k=1}^{m} p(c_j|x_k)^2$$
$$k_j = \arg\max_k p(c_j|k) \tag{4.16}$$

After estimating $p(c_j|B)$, the cross-entropy and parameter updating will be performed as discussed in section 4.2.

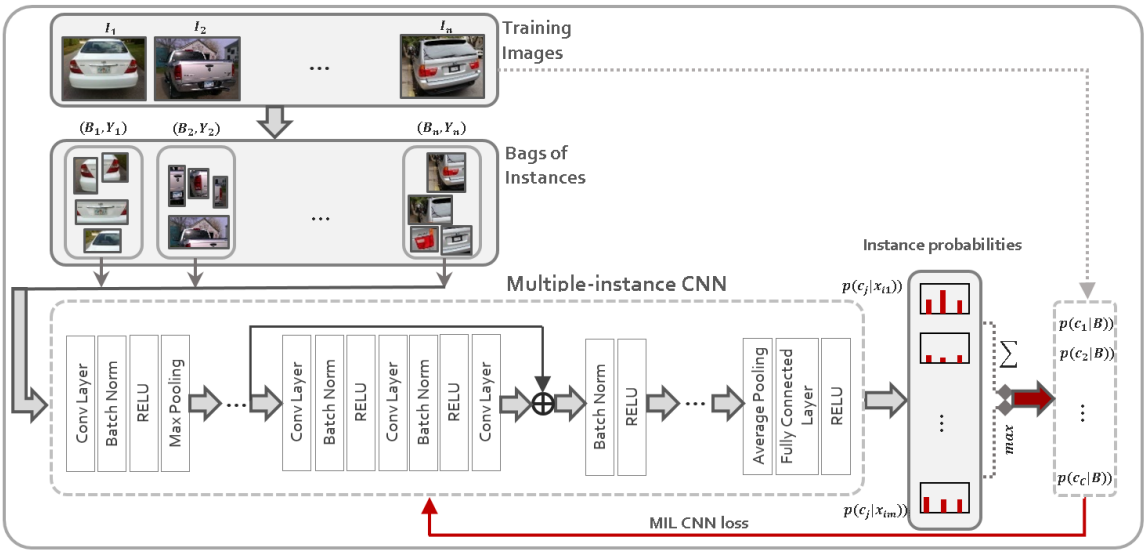


Figure 4.1: Overview of the proposed MIL-based CNN for VMMR

We propose a CNN similar to the ResNet [144], in which convolutional layers mostly have 3×3 filters and down-sampling is directly done by convolutional layers that have a stride of 2. The network ends with a global average pooling layer and a fully-connected layer without softmax. We modify the last layer for MIL. The proposed architecture is illustrated in Figure 4.1.

The following steps outline the general process:

1. Generating randomly cropped regions of each training image;

2. Feeding each set of regions as a bag to the Multiple-instance CNN;

3. Inferring the bag labels based on the network predictions for instances;

4. Training the network with the updated gradients calculated using probability of each bag.

CHAPTER 5

EXPERIMENTAL RESULTS AND DISCUSSIONS

5.1 Dataset Description

Most research efforts on VMMR so far have been focused on medium-scale datasets, which are often defined as datasets that can fit into the memory of a desktop. There are two main reasons for the limited effort on large-scale image based vehicle classification. First, there is almost no publicly available large-scale benchmark data for VMMR. This is mostly because class labels are expensive to obtain. In fact, most existing fine-grained image classification benchmark datasets only consist of a few thousands (or less) of training images. As of existing vehicle datasets, they either cover a subset of makes and models [182,259,260], or only categorize vehicles at a high level (i.e., SUV, Truck, Sedan) [168, 261], and those usable mainly for vehicle detection, pose estimation, and other tasks [53, 200, 262–264]. Second, large-scale classification is hard because it poses more challenges than its medium-scale counterparts. Having the appropriate set of training data can improve the performance of designed classifiers. Indeed, it is necessary to have a very large number of images for each class to cover the wide range of variations of view angles, lighting, as well as the fairly wild appearance difference within the same class. In addition, the large number of vehicles makes, models and years make the number of classes very large.

The lack of public and standard datasets has moved researchers to use their own databases. Accordingly, it is very complicated to establish a performance comparison between the different approaches. A very recently published example is CompCars dataset [21] collected by The Chinese University of Hong Kong. This dataset consists of web-nature and surveillance-nature parts. The former is made of $136,727$ vehicles from 153 car makes with $1,716$ car models, taken from different viewpoints, covering most of the commercial car

models in the recent ten years, and the latter contains 44,481 frontal images of vehicles taken from surveillance cameras. CompCars dataset was originally used for fine-grained car classification, car attribute prediction and car verification. Sochor et al. collected and annotated the BoxCars dataset [202] containing vehicle images taken from surveillance cameras accompanied with their 3D bounding boxes. This dataset is composed of 21,250 vehicles (63,750 images in diverse viewpoints) of 27 different makes, 102 make-model classes, 126 make-model-submodel classes, and 148 make-model-submodel-year classes. Lin et al. [200] published FG3DCar dataset including 300 images of 30 classes. The data provided in FG-Comp [199] includes 8,144 images of cars, covering only 196 makes and models out of which 60% are 2012 models. A vehicle re-identification dataset, VehicleID, collected from multiple real-world surveillance cameras and includes over 200,000 images of about 26,000 vehicles, was introduced in [237]. Almost 90,000 images of 10,319 vehicles in this dataset have been labeled with the vehicle model information.

Despite the ongoing research and practical interests, car make and model analysis only attracts few attentions in the computer vision community. We believe the lack of high quality datasets greatly limits the exploration of the community in this domain. To this end, we collected and organized a large-scale and comprehensive image database called VMMRdb, where each image is labeled with the corresponding make, model and production year of the vehicle. The dataset used in our experiments contains images that were taken by different users, different imaging devices, and multiple view angles, ensuring a wide range of variations to account for various scenarios that could be encountered during testing, in a real-life scenario. The cars are not well aligned, and some images contain irrelevant background. The data was gathered by crawling web pages related to vehicle sales on *craigslist.com*, including 712 areas covering all 412 sub-domains corresponding to US. metro areas. We developed a semi-automated process to partially prune the data and remove the undesired images belonging to interior parts of vehicles.

The VMMR dataset is much larger in scale and diversity compared with the existing car image datasets, containing 9,170 classes consisting of 291,752 images, covering models

manufactured between 1950 to 2016. The distribution of images in different classes of the dataset is illustrated in Figure 5.1. Each circle is associated with a class, and its size represents the number of images in the class. The classes with labels are the ones including more that 100 images.

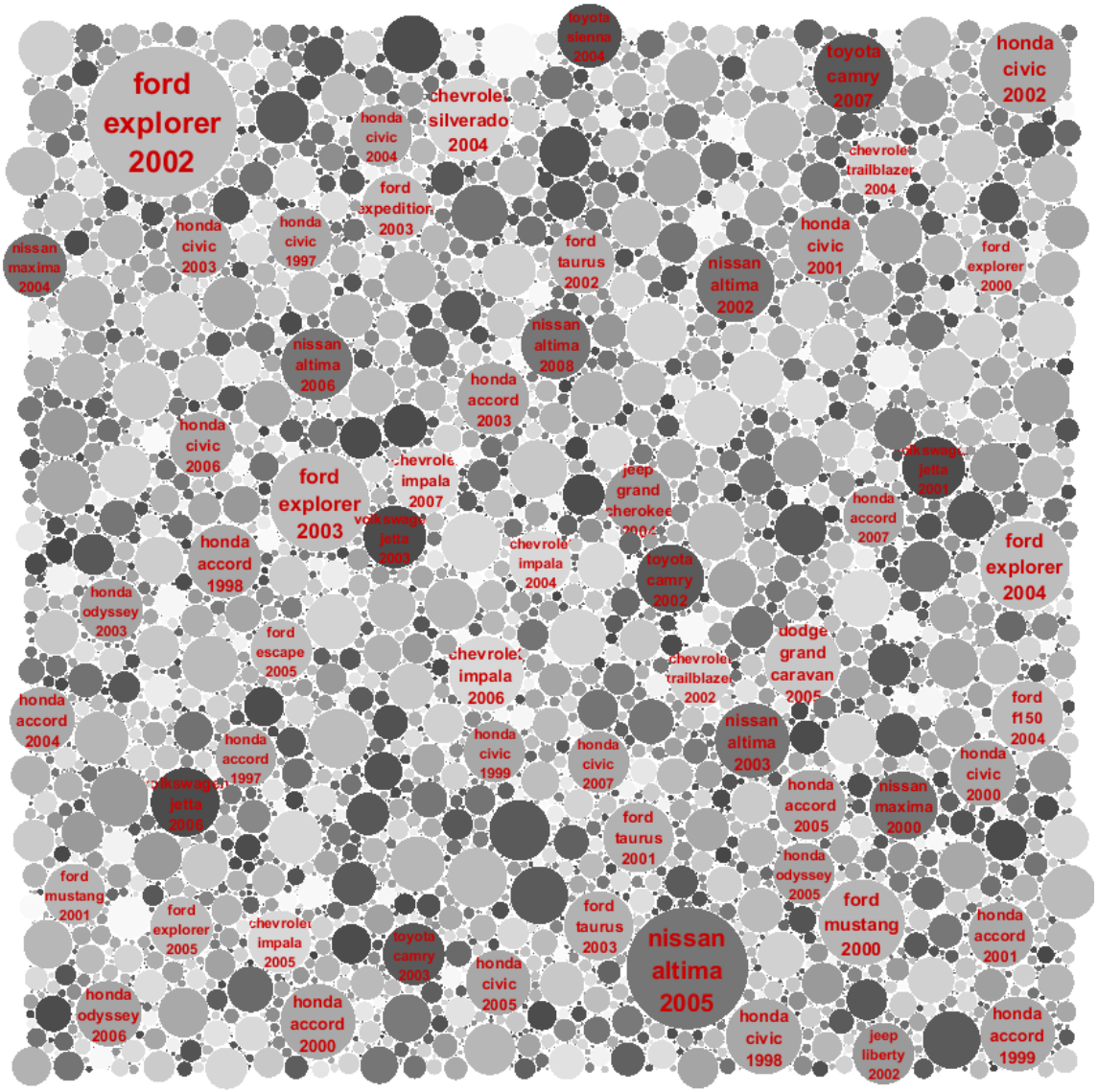


Figure 5.1: Distribution of images per class in VMMRdb

5.1.1 Experimental Subsets

In our experiments, we have used three different versions of our dataset: VMMR-14, VMMR-51 and VMMR-3036. The first dataset, VMMR-14, has 14 classes and for each class, we use 20 images for training and 10 images for testing. For this dataset, we identify regions of interest in two different ways. First, we manually select 6 ROIs within each image. These regions mainly include the tail lights, tailgates and bumpers. We, also, identify another set of ROIs using the saliency detection method as described in section 3.2.1.1.

The second dataset, VMMR-51, is larger. It has 51 classes and for each class we split the images into halves for train and test. These classes are specifically chosen based on the available classes in the CompCars dataset [21] for comparison purposes. In this subset of VMMRdb, we only extract the salient ROIs.

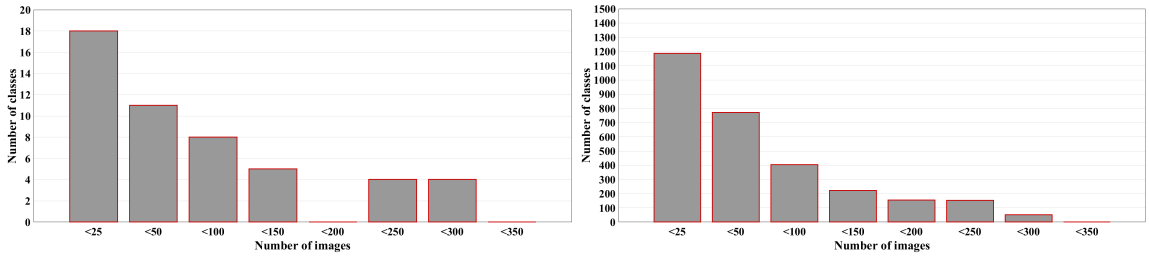
VMMR-3036 is the largest subset which we used for CNN experiments and surveillance application. In this dataset we considered only the classes from VMMRdb with more than 20 images to be able to train the network properly.

Table 5.1 summarizes statics of different versions of dataset that we have used in our experiments. The distribution of the number of images per class is illustrated in Figure 5.2 for the VMMR-51 and VMMR-3036 datasets.

TABLE 5.1

Summary of the VMMR datasets

Dataset	**Number of Classes**	**# Train**	**# Test**
VMMR-14	14	20	10
VMMR-51	51	50%	50%
VMMR-3036	3036	70%	30%



(a) VMMR-51 (b) VMMR-3036

Figure 5.2: Distribution of number of images per class

5.2 Region-based Image Classification

5.2.1 Settings

In our experiments, we have resized all images to 450×600 pixels. All image pixels are represented in the CIE *Lab* color space, which is known for its perceptual uniformity. We assume that the three channels of *Lab* space play equally important roles.

For extracting salient regions, the Boolean maps are sampled by thresholding the color channels only from a distribution over $[0, 255]$. The threshold sample step δ from the color values is set to 8, to create a uniform sampling. An inverted copy of each Boolean map is also considered to justify the inverted region selection. The saliency map has been estimated by applying Gaussian blurring with standard deviation $\sigma = 20$ on the L2-normalized attention maps. The ROIs are then selected by filtering the connected components with the saliency probability and eye-fixation values above 0.3 and 80 respectively. Very large and very small regions are also filtered out. On average, 6 patches were extracted per image.

In all experiments, we follow the same feature extraction process. We sample image patches on a regular spatial grid, with step-size equal to the patch-size, considering only one scale. We compute SIFT descriptors, and in order to decrease the size of feature vector, we reduce the dimensionality from 128 to 80 using PCA, when using FVs to code the appearance.

To model the training data, on the experiments on VMMR-14 we use a mixture of $K = 150$ Gaussian components learned from 1000 samples per each. This process results in feature vectors of size 24000. We increase the number of components to $K = 1000$ for the larger

datasets, and thereby the size of FVs equals 160000.

5.2.2 Verification of Selected Instances

Our first experiment was designed to provide some intuition on the validity of choosing regions which are target of human attention based on their saliency as the only instances in each bag. We asked human annotators to choose certain number of regions in each image as the distinctive parts of the vehicle. In order to keep the complexity consistent, the number of image patches was chosen to be equal to the average value achieved by the automated method. Therefore, for each image we have regions manually annotated with labels from 1 to 6 defining areas representing left and right tail lights, bumper, left and right rear areas of the vehicle including the corresponding tail light and license plate area including the logo respectively. In our proposed saliency-based instance selection process, however, the number of regions per image is not fixed and changes by varying the thresholds applied on saliency and eye-fixation maps. The variance of the salient regions per image with the current setting, is very low. The experiment was performed on VMMR-14 with 6 salient instances selected on average. Figure 5.3 displays the regions selected through each approach for sample images.

We compare the accuracy of the multiple instance (MI) representation when the ROIs are selected manually and when they are selected using the saliency detection algorithm. We also compare the results of the MI representation to those obtained using single instance (SI) representation. For the latter, the entire image is represented by one global feature vector that is extracted in the same way. For the SI learning, we used the standard SVM classifier, and we will refer to this as SI-SVM. For the MI case, we experimented with two MIL algorithms, the instance-level MI-SVM [68] and Citation-kNN [71] (CkNN) classifiers. Let MI-SVM^{m_n} and CkNN^{m_n} refer to the MIL classifier when n ROIs are selected manually, and MI-SVM^{s} and CkNN^{s} refer to the classifier when the ROIs are identified using the saliency detection algorithm. The value of n can vary between 2 and 6.

In order to see the effect of varying the number of selected instances on the performance of

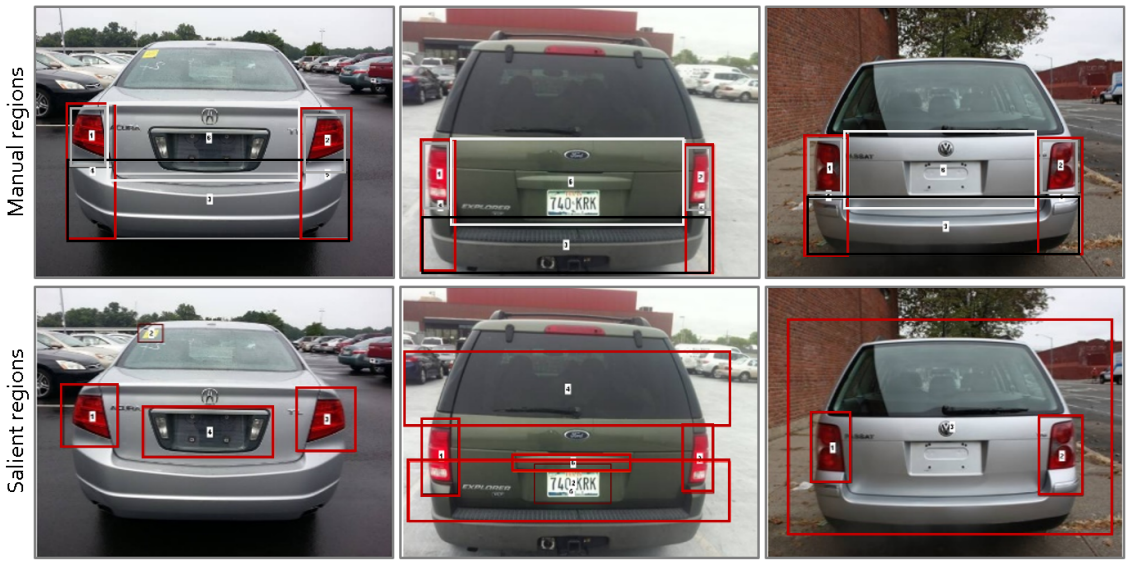


Figure 5.3: Examples of regions selected through manual annotation and saliency detection for sample images from VMMR-14

MI-classifier, we chose different subsets of the annotated regions. The results are depicted in Figure 5.4. As we can see, certain sets of regions are more discriminative in certain classes. However, on average the accuracy does not change a lot by increasing the number of instances. Therefore, in the rest of our experiment, to keep the complexity consistent with our proposed instance selection process, we have employed all 6 manually selected regions. For the sake of simplicity, from here on, we are going to refer to MI-SVM^{m_6} and CkNN^{m_6} with MI-SVM$^{\mathrm{m}}$ and CkNN$^{\mathrm{m}}$ respectively.

Figure 5.5 compares the results of SI-SVM, MI-SVM$^{\mathrm{m}}$ and MI-SVM$^{\mathrm{s}}$ on the VMMR-14 data. As it can be seen, the performance of our automatic saliency-based method is comparable to the method with manually selected instances. The accuracy of both experiments, however, are significantly better than the case where the model has been trained for one global instance per image.

5.2.3 Comparison of SI and MI Learners

In a second experiment, we compare the performance of SI-SVM with MI-SVM$^{\mathrm{s}}$, and CkNN$^{\mathrm{s}}$ on VMMR-14 and VMMR-51 datasets. The results are depicted in Figure 5.6 for

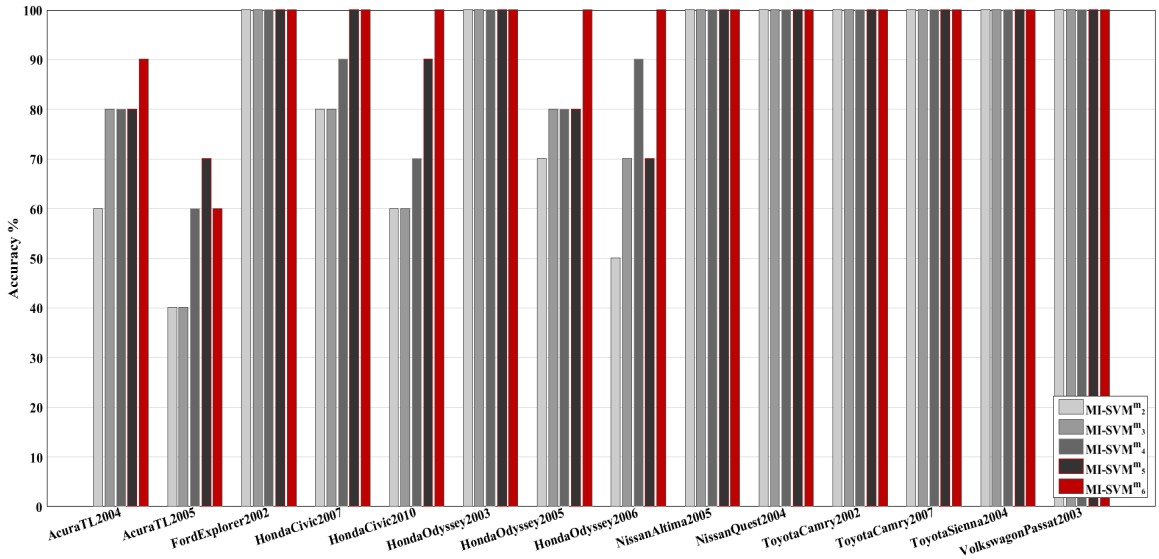


Figure 5.4: Classification results using MI-SVM with different subsets of manually selected regions (MI-SVM^{m}) on VMMR-14

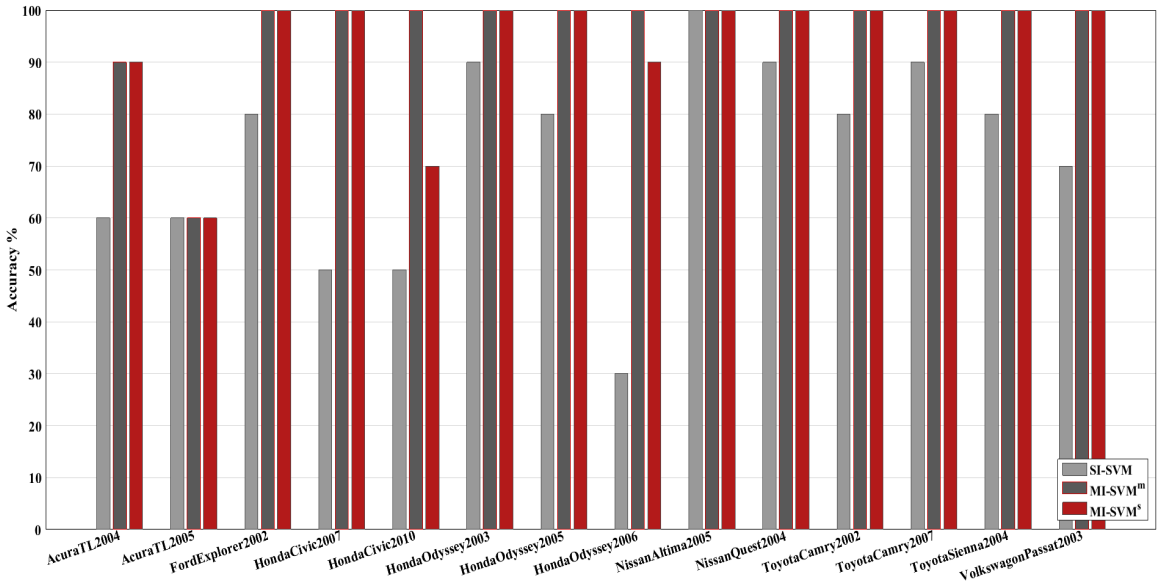


Figure 5.5: Classification results using single instance learning (SI-SVM), MI-SVM with manually selected regions (MI-SVM^{m}) and MI-SVM with ROIs selected using saliency detection algorithm (MI-SVM^{s}) on VMMR-14

VMMR-14. As it can be seen, the MI representation outperforms the SI representation for most of the classes. The improvement in accuracy can exceed 40% in some cases. The MI classifiers have comparable accuracies for all classes. This indicates that the improvement

in accuracy is due mainly to the MI representation and not the specific MIL algorithm. Considering the slightly better performance of MI-SVM we have used this classifier for the experiments on the larger datasets.

Table 5.2 summarizes the classification accuracy of the discussed experiments. The confusion matrix of the performance of VMMR-51 with respect to SI-SVM is depicted in Figure 5.7.

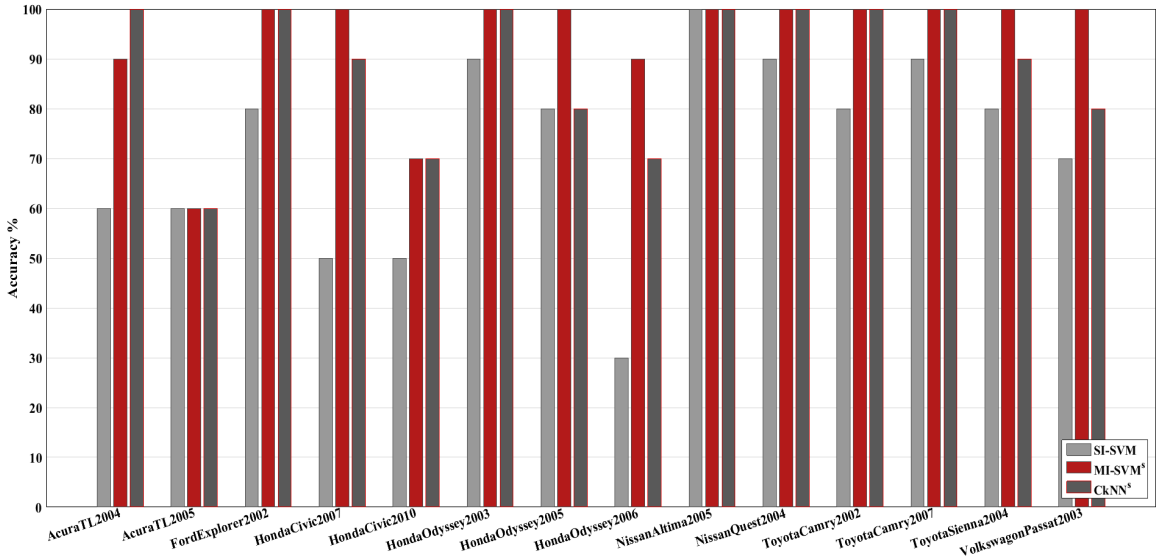


Figure 5.6: Classification results using single instance learning (SI-SVM), MI-SVM (MI-SVM^s) and CkNN (CkNN^s) with ROIs selected using saliency detection algorithm on VMMR-14

TABLE 5.2

Classification Accuracy(%) of SI vs. MI Experiments

Dataset	**SI-SVM**	**MI-SVM^m**	**MI-SVM^s**	**CkNN^s**
VMMR-14	72.14	96.43	93.57	88.57
VMMR-51	37.53	N/A	73.81	N/A

Figure 5.8 displays three sample bags from various classes misclassified by SI-SVM, but classified correctly using MI-SVM and CkNN. Each image includes a subset of the ROIs representing the instances with label probability greater than 0.4. The red and white

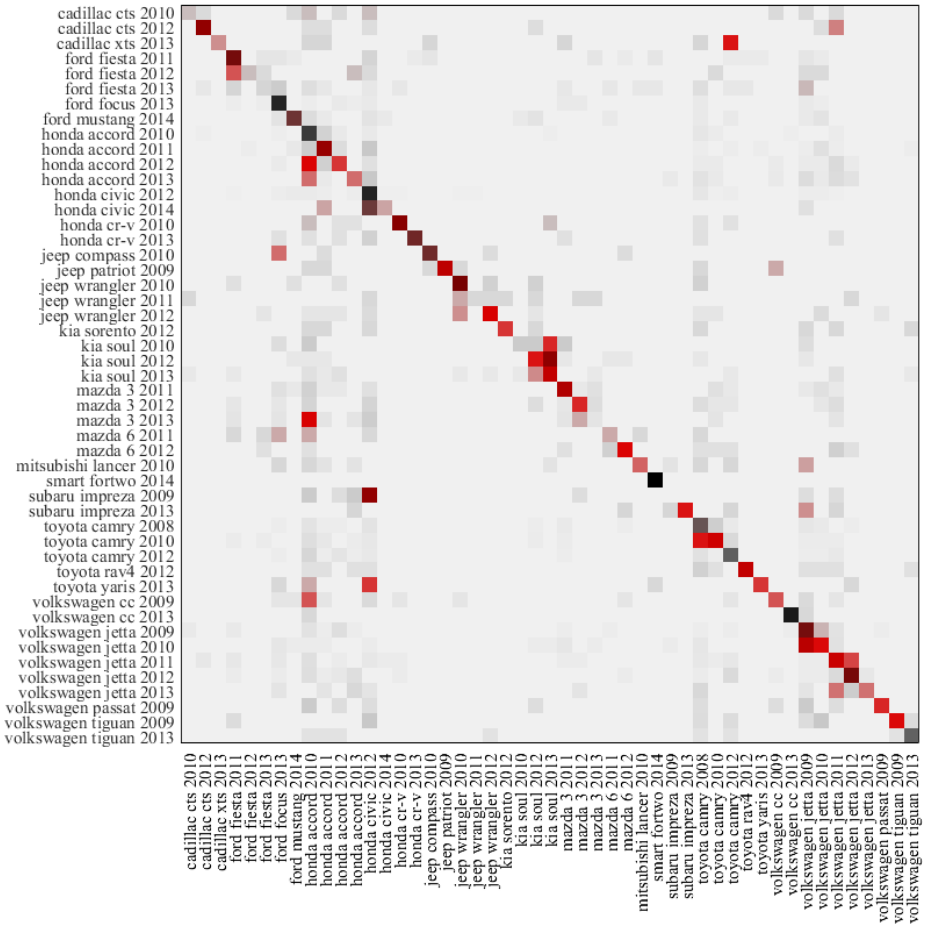


Figure 5.7: Confusion matrix of SI-SVM on VMMR-51

regions identify the instances classified correctly and incorrectly, respectively. The number displayed on top of each bounding box identifies the corresponding instance classification probability. As we can see, the missclassified examples of SI are mainly suffering from circumstances such as poor lighting, partial occlusion, open trunk, etc. For instance, the image including BMW X5 with open trunk was a missclassified item in SI where it was classified correctly using MI-SVM since one of its instances (tail light ROI) was assigned a high probability.

5.2.4 Analysis of Discriminative Regions

In this part of experiments, we would like to address an essential question of MIL: *Which instances indeed contribute to the semantic meaning of the bag-level labels?*

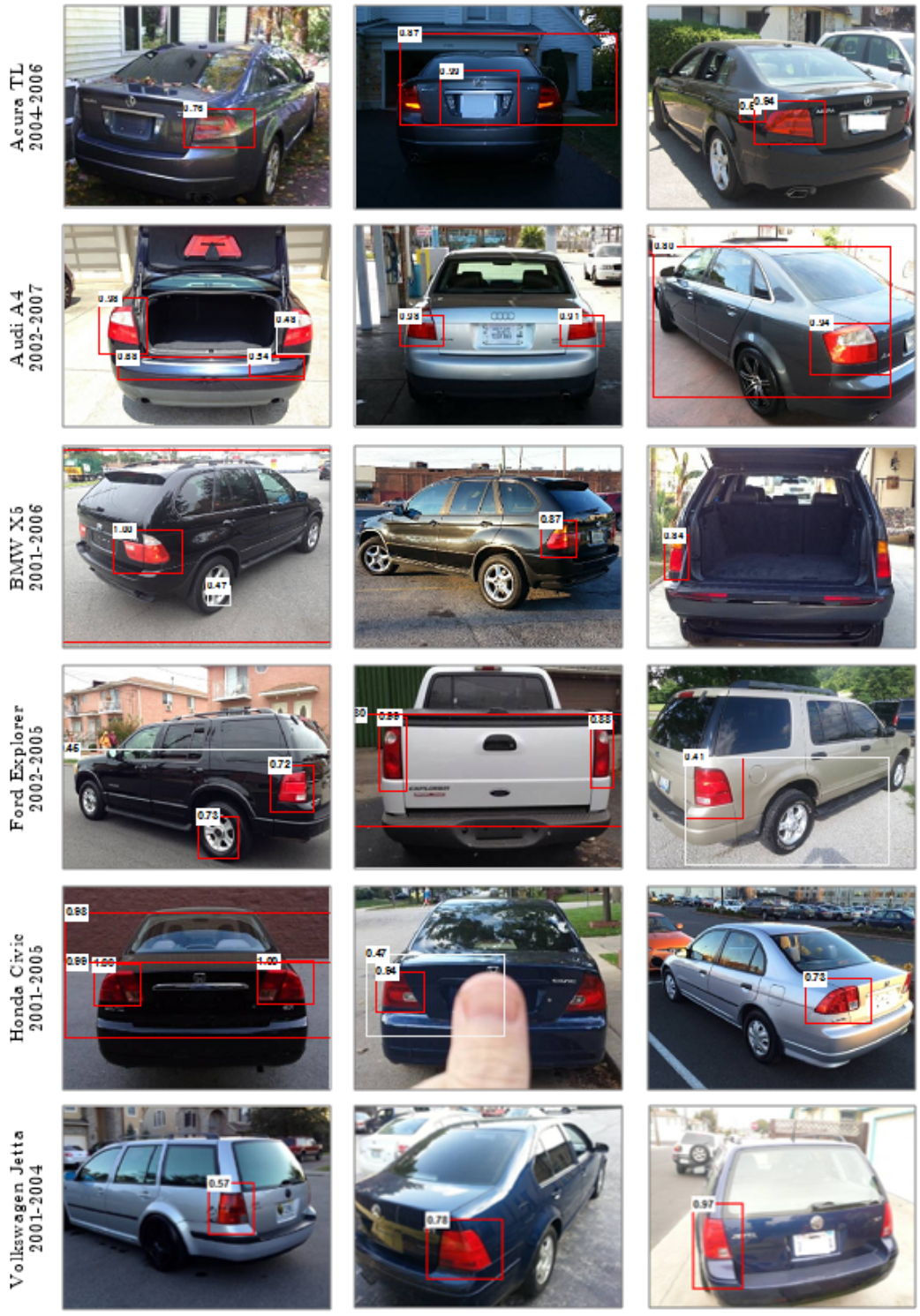


Figure 5.8: Sample images misclassified by SI-SVM, but correctly classified using MI-SVM. For each image we show the ROIs that have high probability in the class under consideration.

Although the instances based on the saliency map and eye-fixation prediction are supposed to represent the most distinctive parts of the object, we performed another study to interpret the selected instances and see if the top regions vary per class. In other words, we wanted to find the most semantic instances in each make and model. In Figure 5.9 the instances with the highest probability in some sample classes of VMMR-14 are shown. It is clear that the back lights in almost all of the classes are always among the top regions. In certain classes such as Ford and Volkswagen, the make logo is listed among the top instances due to their visually different shape. Another interesting interpretation is that depending on the viewpoint, the distinctive instances vary. This proves another advantage of the proposed method over existing solutions for VMMR, which makes it robust to view-point changes.

Figure 5.9: Top 5 instances retrieved for sample classes from VMMR-14

5.3 Impact of Diverse Data

In the next experiment, we analyze the importance of having diversity in data to handle real-world surveillance applications. We commence our evaluations by comparing the performance of our dataset with the CompCars dataset employing the same settings as the ones used in [21], using CNN learners.

Despite having hierrarchical labels of make, model and year in their datase, Yang et al. [21] have merged all production years of each model to the same class in their experiments. This has resulted in 431 classes, many of which are chinese manufacturers. To have a proper comparison, we choose only those classes existing in our dataset (125 classes) and following their approach, we use the labels at the make-model level only. We pick the exact year for which any image is included in CompCars. In the resulting 51 classes we divide the images into 50% for training and 50% for testing. Table 5.3 details the number of images in each dataset.

TABLE 5.3

Specifications of the overlap data between CompCars and VMMR datasets

Dataset	Number of Classes	# Train	# Test
CompCars-51	51	1527	1506
VMMR-51	51	1986	1984

5.3.1 Model Perspective

We analyzed the effect of network architecture, comparing VGG and ResNet considering their state of the art performance in several fine-grained classification problems (refer to section 2.4.2). In all cases, using a ResNet significantly improves results, so we present the remainder of the experiments using a ResNet for feature extraction. In this section, we compare the recognition performance of each datset on Resnet-50. Our baseline convolutional neural network consists of a Conv-ReLU-Pool set, followed by a fully-connected layer with softmax loss.

Despite using modern GPUs, training a full CNN from scratch for a large scale problem can take a week or two, which is too slow for many research areas. Moreover, training a CNN of a similar size requires a tremendous amount of training data, which is not available for many tasks, where data collection poses a challenge itself. Several works have experimented with using the outputs of a pre-trained CNN as an image/patch descriptor for a new task other than the data the CNN was trained on. It is assumed that the bottom layers, especially the convolutional layers, correspond to generic image representations while the top layers are task-specific. Therefore, in our experiments we use networks pre-trained on ImageNet, and fine-tuned on the datasets under experiment with the same mini-batch size, epochs, and learning rates.

There are several software platforms, where training and classification of a deep neural network are straightforward engineering tasks, such as Torch [265], Theano [266], and Caffe [267], to name a few. In our work, we used Torch to construct, train, and test our networks.

We investigate the classification accuracy of networks trained on each dataset in confronting with samples from other datsets. The nature of images provided in CompCars are very different from VMMRdb images in the sense that they are mostly captured in more controlled environment with much higher resolution. The purpose of these experiments is to see how well the model performs given images collected in more challenging scenarios. We, also, generate a third dataset which we refer to as CompCarVMMR-51, by merging the discussed datasets. The performances of these experiments are summarized in Table 5.4. We report the "Top-1" and "Top-3" accuracies of car make-model classification, which denote the classification accuracy considering the first and up to three top matches, respectively, for each pair of train and test set.

As we expected, the model trained on CompCars, despite its significant performance on the test set with images of similar nature, degardes considerably on the test images selected from VMMR-51. The performance of the model trained on VMMR-51 is just slightly better with respect to non-VMMR images. The merged dataset, however, outperforms

TABLE 5.4

Classification results for the models trained on different datasets

Train	Test			
	CompCars-51	**VMMR-51**	**CompCarVMMR-51**	
CompCars-51	96.88	36.10	62.23	*Top-1*
	97.88	50.05	70.69	*Top-3*
VMMR-51	40.28	90.26	68.22	*Top-1*
	52.85	93.48	75.93	*Top-3*
CompCarVMMR-51	96.61	94.10	95.16	*Top-1*
	97.48	96.47	96.91	*Top-3*

both previous cases, proving the fact that employing additional training data can boost classification results by increasing data diversity in training examples.

5.4 Fine-grained VMMR

We extended the CNN-based experiments to train a model for classification of vehicles make, model and production year on VMMMR-3036. We set the parameters to learning rate 0.01, and 200 epochs. The learning rate decay was applied after initial 30 epochs. In training, all inputs were color-normalized with the mean and standard deviation from ImageNet images after scale, aspect ratio, color, and horizontal flip augmentations. During the test phase, all detections were center-croped and color-normalized by the system. We trained the models with a minibatch size of 32 within 110 hours on a NVIDIA GeForce GTX 1080 GPU using ~8 Gb of memory. Table 5.5 lists the accuracy of different deep architectures. Considering the superior validation accuracy achieved by ResNet-50 with respect to having less parameters, we choose this model for the AVS application.

Figure 5.10 illustrates some predictions, indicating that the model accounts for data variations in different viewpoints and lighting conditions. Each image is represented with its ground truth label along with the top-5 probabilities. The images in the first and second

row display samples correctly and incorrectly classified by the ResNet-50 model, respectively. As we can see, many of the misclassified examples have been categorized as the same and model as ground truth, but different year. Figure 5.11 displays the vehicle images that trigger high responses with respect to each neuron in the last fully-connected layer of the model trained with ResNet-50 architecture. As we can see, different neurons capture car images of specific car models across different viewpoints; showing that CNN model is capable of learning discriminative representation across different views.

To observe the learned feature space of the model, we projected the features extracted from the last fully connected layer to a two-dimensional embedding space using t-Distributed Stochastic Neighbor Embedding (t-SNE) [268]. The projections are displayed in Figure 5.12 for few images from sample classes. We can see that features from the same model are closer to each other compared to the ones visually very different.

TABLE 5.5

The classification accuracies of different deep models on VMMR-3036

Model	**VGG**	**ResNet-50**	**ResNet-101**
Top-1	44.39	51.76	50.74
Top-5	91.88	92.90	93.07

5.4.1 Multiple-Instance CNN

In our implementation of the model represented in 4.1, we extracted 24 instances per image by randomly cropping each input bag. We used the Resnet-50 model pretrained on CompCars-51 and fine-tuned it on CompCarsVMMR-51. Table 5.6 summarizes the performance of the model.

5.5 Vehicular Surveillance

Our proposed methods can offer valuable situational information for law enforcement units in a variety of civil infrastructures. Because of the natural environments and uncon-

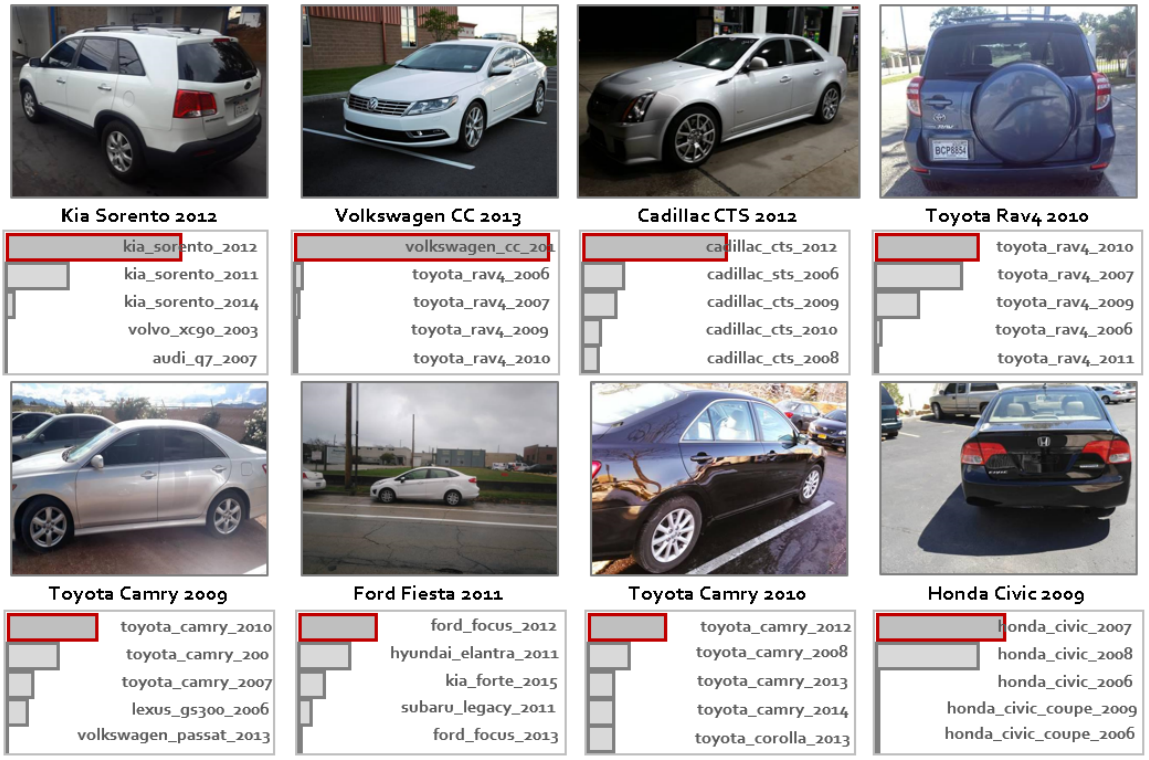


Figure 5.10: Top-5 predicted classes of the CNN model for sample images from VMMR-3036. Below each image is the ground truth class and the probabilities for the top-5 predictions with the matched class in the top bar.

TABLE 5.6

Performance comparison of proposed approaches on the CompCarVMMR-51 dataset

	MIL	**CNN**	**MIL-CNN**
Top-1	72.16	95.16	88.16
Top-5	N/A	99.12	95.27

strained image settings, our dataset can be used as a baseline for training a robust model. In this section, we conduct a cross-modality experiment, where the CNN model fine-tuned by the web-nature data is evaluated on the surveillance video streams. To demonstrate the effectiveness of our proposed approaches for VMMR, we target an important real-life surveillance application where our system would be able to analyze video data acquired from multiple surveillance cameras to monitor vehicles under varying environment and capture

Figure 5.11: Images with the highest response from sample neurons of the FC layer of ResNet-50. Each row corresponds to a specific neuron.

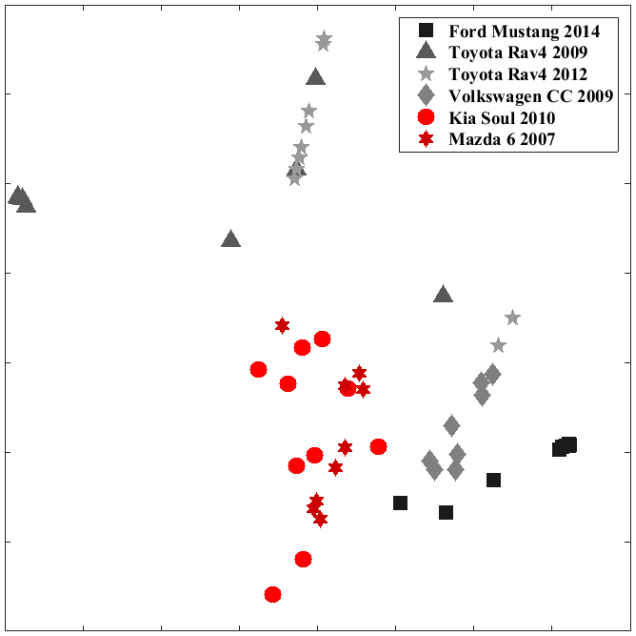


Figure 5.12: Features of sample car models projected to a 2D embedding using multi-dimensional scaling

conditions.

The benefits are multi-fold: Cameras are prevalent and cost-efficient tools for moni-

toring, and they often have other surveillance uses simultaneously. Using the data acquired this way, our system could also search for a target vehicle in the CCTV feeds and perform vehicle re-identification, in case of security related investigations, using only the meta information extracted while detection. A reliable traffic monitoring system would not only make the roads safer but can also potentially disrupt criminal activities. Most technology solutions for vehicular monitoring involve significant human supervision or are passive forensic tools. These systems allow many cameras to be observed by a small number of trained human operators but suffer from potential operator fatigue and lack of attention due to the large amount of information provided by cameras which can distract the operator from focusing on important events. Therefore, automated methods are needed to screen traffic in a non-obstructive manner and detect anomalies. Additionally, the data gathered in a certain period of time can be analyzed further for traffic evaluations or marketing purposes.

5.5.1 Target Environment

In our experiments, the cameras are fixed and the input to the algorithm would typically consist of rear view images, such as those shown in Figure 5.13. In countries such as the USA, it is not usually required for drivers to display license plates on both ends of their vehicles. In fact, in most states, it is common to display license plates only on the rear-end of vehicles. Given this insight, video traffic surveillance systems are typically deployed in a way that allows capture of license plates. The focal length is fixed to maximize the size of the rear part of each vehicle projected into the image plane, assuring that large vehicles are totally visible. Figure 5.14 depicts the position of the cameras on a map in our experimental setup. Vehicles may be occluded by pedestrians or other objects (images in the second row of Figure 5.13). The images were captured in one day under different lighting conditions.

Figure 5.13: Sample frames of the video footages used in the surveillance experiment

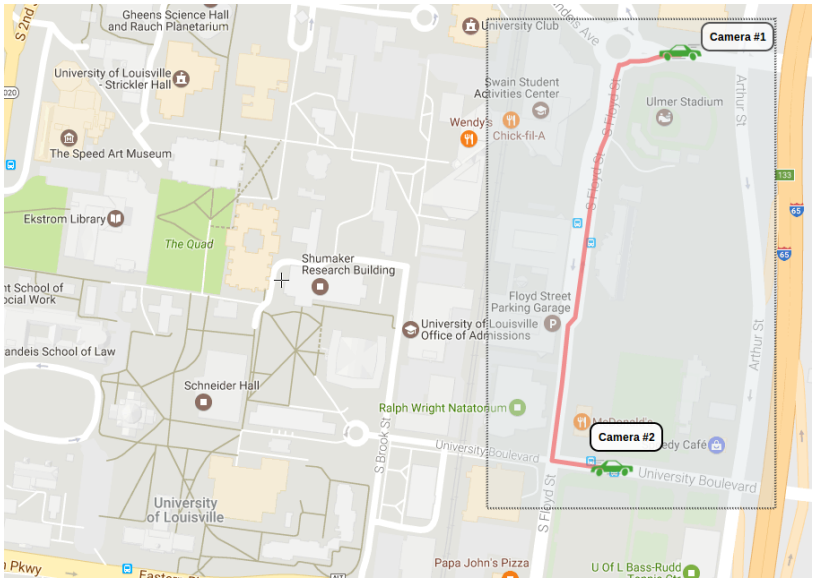


Figure 5.14: Camera positions in the vehicle surveillance experiment

5.5.2 Vehicle Re-Identification

The pipeline of our vehicular surveillance application is illustrated in Figure 5.15. We deploy the model trained on the VMMR dataset (section 5.4) in the aforementioned AVS system to detect, track and identify vehicles.

In our implementation, the testing time is 52 FPS for vehicle boundingbox detection and 112 FPS for make/model/year recognition.

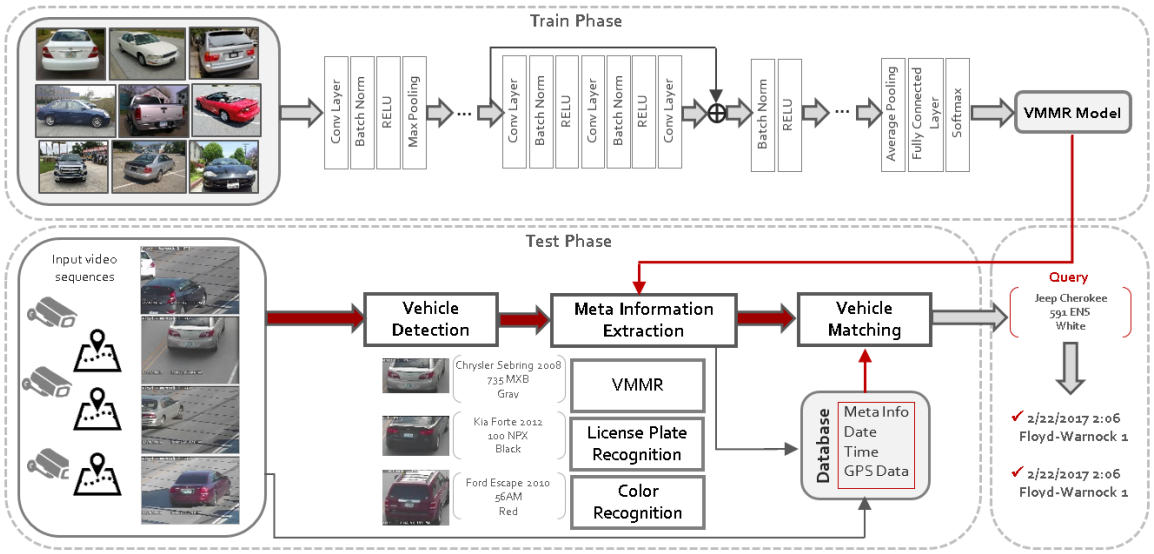


Figure 5.15: AVS pipeline

5.5.2.1 Vehicle Detection

In our experiments, we extracted bounding boxes using YOLOv2 [269] trained on train and validation sets of MS COCO dataset [270]. Out of the existing 80 classes in the MS COCO dataset, we used the model to extract the regions only containing classes corresponding to cars, trucks and buses. Separating the vehicle detection and VMMR blocks allows us to considerably reduce the VMMR complexity and modify the classification module according to the desired classification goals.

5.5.2.2 Meta Information Extraction

A typical access control setup extracts individual frames from a video stream for vehicle identification. We compute the likelihoods for each car make and model for all the frames where a vehicle is detected. A set of frames are associated with a vehicle based on the overlap ratio of the bounding boxes in the consecutive frames and the variation of its size with respect to its speed. Computing the average (or maximum) of these likelihoods per vehicle, tends to improve the accuracy of recognition, since a larger number of frames has a higher probability of a good rear view of the vehicle (avoiding e.g, cases with car shown only partially, blocked by another vehicle or changing the lane).

Additionally, we extract the license plate, if available, as the second part of the vehicle identification stage. In order to localize the plate, the vertical edges are first detected to generate edge density map, and then binarization and dilation are performed on such a map, and finally the license plate will be located through connected component analysis. In this phase we take advantage of the prior information on the approximate location and size of the license plate with respect to the vehicle's bounding box.

Another block of the feature extraction process is assigned for recognition of vehicle color which is an important verification property and provides visual cues to boost the system's accuracy. It is a very challenging task due to several factors such as weather condition, illumination variation, vehicle speed, and difficulty in discriminating between certain colors in uncontrolled environments. To reduce the over exposure problem, we convert the image to other color spaces that separate illumination and color, such as CIE Lab or HSV. The vehicle color is classified based on the distances of the values of multiple color spaces between the region of interest (ROI) and those of training samples.

Figure 5.16 illustrates the extracted information from sample frames containing vehicles in two different cameras used in our experimental setup. Given the information of target vehicle in camera #1, we search for the matching candidates in the second camera in the next phase of algorithm. Top-3 candidates with their corresponding meta-information are displayed in the second row of Figure 5.16.

5.5.2.3 Vehicle Matching

With the recorded meta information of each vehicle, at the time of retrieval, we search for all the possible matches in a specific time interval. We match the license plate against the car make, model and color, and track the target vehicle between multiple cameras. We assign different weights to the dominant make and model, color and similarity of license plate matching for the matched vehicles. Figure 5.17 shows some sample frames with vehicles correctly matched between two cameras. The first row represent the target vehicle and the next 3 rows display the top-3 matches for the target vehicle in the second camera along

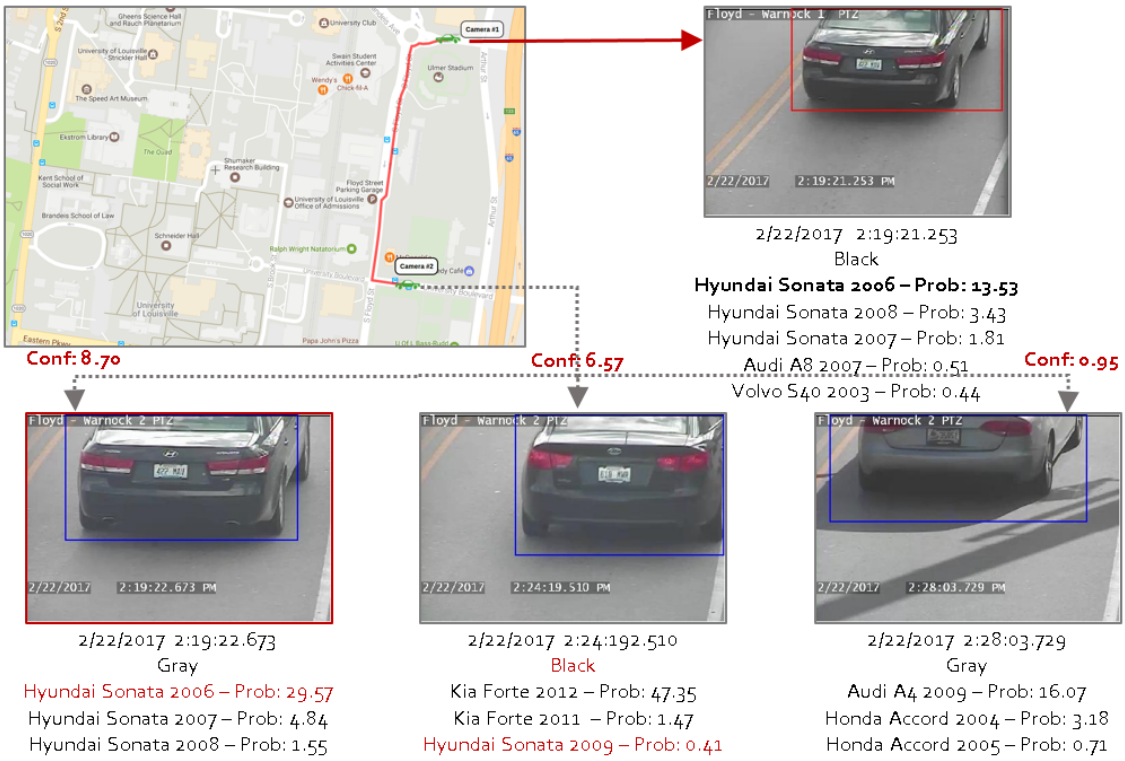


Figure 5.16: Meta-information extraction for sample frames captured in two different cameras. The top image shows the current frame in camera #1 on the map, and the images in the second row represent candidate matching frames from camera #2.

with its confidence value calculated by the algorithm.

The system, however, matches the target vehicle with an incorrect vehicle in some cases due to several changes such as blurred license plate, illumination changes affecting vehicle color, and incorrect labeling of VMMR system. Figure 5.18 depicts some of these examples. We can observe that in most of these cases the incorrect matches are visually very similar. The system, however, assigns much lower confidence value to these cases compared to the samples from Figure 5.17.

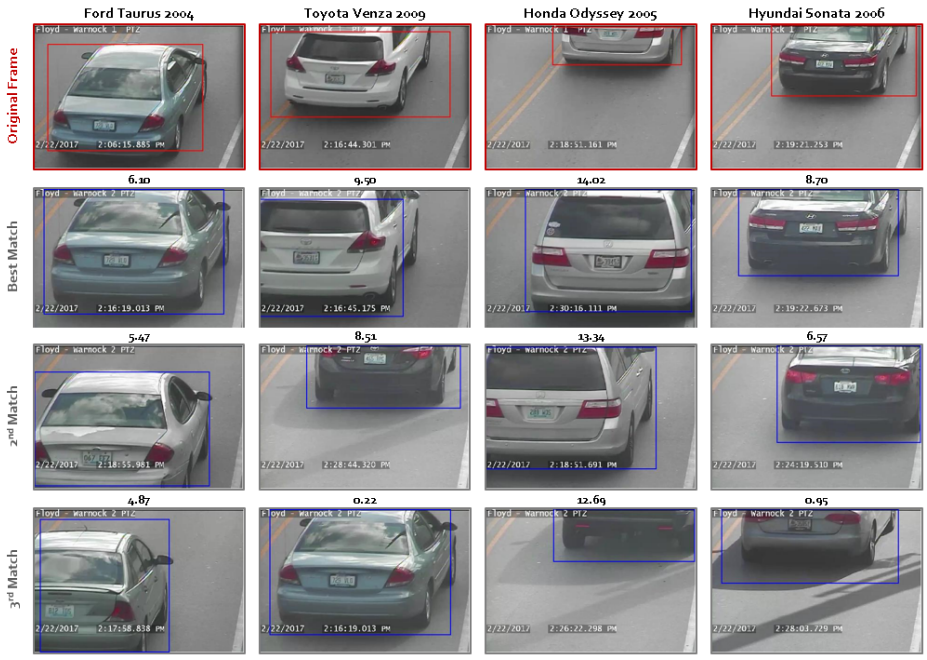


Figure 5.17: Sample vehicles detected in the first camera matched with the vehicles from the second camera. Below each frame top-3 matches are displayed with their corresponding confidence value.

Figure 5.18: Sample vehicles detected in the first camera, incorrectly matched with the vehicles from the second camera.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In this dissertation two schemes for fine-grained vehicle classification are studied and evaluated to address the multiplicity and ambiguity problems of VMMR.

Traditionally, fine-grained classification methods rely on detailed manual annotations which are intrinsically ambiguous, time-consuming and very costly even for average-scale real-world datasets. An alternative is to learn local concepts using global annotations. This approach is specially designed for fine-grained categorization in weakly-supervised scenario, because distinctive parts have been shown to play an important role in the existing annotation dependent works.

We have introduced an MIL-based framework for VMMR to categorize fine-grained images without using any object/part annotation either in the training or in the testing stage. In multiple instance problems, instances are grouped into bags, labels of bags are known but not those of individual instances. In this scenario, a bag is a full size image and an instance is a distinguishable patch. The bag is labeled positive if and only if there is at least one positive instance in the bag, i.e. some part of the image, but maybe not the whole image include a vehicle. In our approach, the high-dimensional feature space generated by having several random instances extracted from the image in traditional methods, is considerably shrunk by filtering the instances using a Boolean map-based saliency model. By exploiting the surroundedness cue for eye fixation prediction, we prune the instances to more semantic ROIs, while keeping a reasonable number of instances per bag. The resultant salient patches in each image are presented to MIL by extracting FV which has proven to be very effective in image classification.

We, also, incorporated deep learning into a weakly supervised learning framework to enhance the feature representation in the first approach while taking advantage of distinct part discovery to accomplish high-level tasks such as VMMR. We modified the architecture and the loss function of CNN to make it adaptable to the MIL settings and fed randomly cropped patched of each input image as instances to the network.

We collected a comprehensive dataset, VMMRdb, to help experiments in this direction by providing sufficient amount of data enriched by information automatically extracted to define each vehicle'make, model and production year. The effectiveness and superiority of our approaches over the baseline classifiers are validated on different subsets of this dataset. In experiments, the proposed approach consistently outperforms single instance-based classifiers. The results are also comparable to the instances extracted by human annotators from distinctive regions. Our experiments prove the idea that one of the key points for fine-grained categorization is to find the most distinctive parts describing the object. Using those patches for categorization is also able to save the computation complexity, especially in the case of large number of categories.

As a drawback of the deep neural network, however, there is a need to define the hyper-parameters of the network. The choices of the number of layers, number of nodes, sizes of convolutional kernels, etc. all have a crucial importance on the resulting accuracy.

The presented methods have the potential to improve the functionality of current traffic camera systems. We, as humans, are also not very good at reading cars license plates unless they are quite near us, nor are we very good at remembering all the characters. However, we are good at identifying and remembering the appearance of cars, and therefore their makes and models, even when they are speeding away from us.

Thus, we organized our system in form of a real-life application with surveillance and security purposes. We stored the results of our VMMR system estimated over a period of time for the footage recorded from security cameras mounted around the campus. This data can be beneficial for both transportation as well as businesses authorities.

6.2 Potential Future Work

Despite the significant progress that have been made in vehicle surveillance during the recent years, many challenging issues still need further research and development especially in urban traffic scenarios such as road sections and intersection in which dense traffic, vehicle occlusion, pose and orientation variation and camera placement highly affect their performance. In road sections vehicles usually travel in a uni-direction in which heavy traffic and congestion may affect vehicle detection due to slow or temporary stopped vehicles. Vehicle's pose and orientation with respect to the camera often varies while moving within intersections due to lane change and turn left, right and round. This will vary the appearance and scale of vehicle within consecutive frames affecting tracking and classification dramatically. Nighttime is, also, a dramatic challenge for traffic surveillance, in which headlight and taillights are used to represent the vehicle.

Additionally, the proposed approaches for VMMR can be easily applied to other scenarios in which the camera is not fixed, e.g., an on-board camera on a mobile surveillance vehicle, etc.

In another direction, enhancing the process of logo recognition can considerably contribute to the performance of VMMR system. A reliable logo detection algorithm must not only meet the challenge of variations in a logo's visual appearance but also distinguish logos from other small visual patterns that may appear on the vehicles. Although the classification of vehicle brand by using logo information has been a subject of interest for several years, the bottleneck of logo detection has remained largely unsolved.

REFERENCES

[1] Why Superior Network Performance Matters, "A focus on efficiency," 2013. [Cited on page 1.]

[2] YouTube, "Youtube press statistics," https://www.youtube.com/yt/press/statistics.html/, 2015 (accessed April 1, 2016). [Cited on page 1.]

[3] Ryan Farrell, Om Oza, Ning Zhang, Vlad I Morariu, Trevor Darrell, and Larry S Davis, "Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 161–168. [Cited on pages 1, 7, and 33.]

[4] Bangpeng Yao, Gary Bradski, and Li Fei-Fei, "A codebook-free and annotation-free approach for fine-grained image categorization," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3466–3473. [Cited on pages 1, 17, and 79.]

[5] Ning Zhang, Ryan Farrell, and Trever Darrell, "Pose pooling kernels for sub-category recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3665–3672. [Cited on page 1.]

[6] Maria-Elena Nilsback and Andrew Zisserman, "A visual vocabulary for flower classification," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. IEEE, 2006, vol. 2, pp. 1447–1454. [Cited on page 1.]

[7] Maria-Elena Nilsback and Andrew Zisserman, "Automated flower classification over a large number of classes," in *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*. IEEE, 2008, pp. 722–729. [Cited on page 1.]

[8] Neeraj Kumar, Peter N Belhumeur, Arijit Biswas, David W Jacobs, W John Kress, Ida C Lopez, and João VB Soares, "Leafsnap: A computer vision system for automatic plant species identification," in *Computer Vision–ECCV 2012*, pp. 502–516. Springer, 2012. [Cited on page 1.]

[9] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li, "Novel dataset for fine-grained image categorization: Stanford dogs," in *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, 2011. [Cited on page 1.]

[10] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi, "Fine-grained visual classification of aircraft," *arXiv preprint arXiv:1306.5151*, 2013. [Cited on page 1.]

[11] Eleanor Rosch, Carolyn B Mervis, Wayne D Gray, David M Johnson, and Penny Boyes-Braem, "Basic objects in natural categories," *Cognitive psychology*, vol. 8, no. 3, pp. 382–439, 1976. [Cited on pages 2 and 7.]

[12] Statistica, "The statistics portal," https://www.statista.com/statistics/281134/number-of-vehicles-in-use-worldwide/, 2014 (accessed February 1, 2017). [Cited on page 3.]

[13] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788. [Cited on page 3.]

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *European Conference on Computer Vision*. Springer, 2014, pp. 346–361. [Cited on page 3.]

[15] Noppakun Boonsim and Simant Prakoonwit, "Car make and model recognition under limited lighting conditions at night," *Pattern Analysis and Applications*, pp. 1–13, 2016. [Cited on page 3.]

[16] Edward Hsiao, Sudipta N Sinha, Krishnan Ramnath, Simon Baker, Larry Zitnick, and Richard Szeliski, "Car make and model recognition using 3d curve alignment," in *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*. IEEE, 2014, pp. 1–1. [Cited on pages 3, 53, and 60.]

[17] V Kastrinaki, Michalis Zervakis, and Kostas Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image and vision computing*, vol. 21, no. 4, pp. 359–381, 2003. [Cited on page 3.]

[18] Christos-Nikolaos E Anagnostopoulos, Ioannis E Anagnostopoulos, Ioannis D Psoroulas, Vassili Loumos, and Eleftherios Kayafas, "License plate recognition from still images and video sequences: A survey," *IEEE Transactions on intelligent transportation systems*, vol. 9, no. 3, pp. 377–391, 2008. [Cited on page 4.]

[19] Ying Wen, Yue Lu, Jingqi Yan, Zhenyu Zhou, Karen M von Deneen, and Pengfei Shi, "An algorithm for license plate recognition applied to intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 830–845, 2011. [Cited on page 4.]

[20] Shan Du, Mahmoud Ibrahim, Mohamed Shehata, and Wael Badawy, "Automatic license plate recognition (alpr): A state-of-the-art review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 311–325, 2013. [Cited on page 4.]

[21] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang, "A large-scale car dataset for fine-grained categorization and verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3973–3981. [Cited on pages 4, 47, 56, 86, 89, and 98.]

[22] Jun-Wei Hsieh, Li-Chih Chen, and Duan-Yu Chen, "Symmetrical surf and its applications to vehicle detection and vehicle make and model recognition," *IEEE Transactions on intelligent transportation systems*, vol. 15, no. 1, pp. 6–20, 2014. [Cited on pages 5 and 60.]

[23] Catherine Wah, Steve Branson, Pietro Perona, and Serge Belongie, "Multiclass recognition and part localization with humans in the loop," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2524–2531. [Cited on page 7.]

[24] Jeff Donahue and Kristen Grauman, "Annotator rationales for visual recognition," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1395–1402. [Cited on page 8.]

[25] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng, "Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks,"

in *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2008, pp. 254–263. [Cited on page 8.]

[26] Richard O Duda, Peter E Hart, and David G Stork, *Pattern classification*, John Wiley & Sons, 2012. [Cited on page 9.]

[27] Dengxin Dai and Luc Van Gool, "Ensemble projection for semi-supervised image classification," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2072–2079. [Cited on page 9.]

[28] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial intelligence*, vol. 89, no. 1, pp. 31–71, 1997. [Cited on pages 9, 10, 18, 19, 20, and 23.]

[29] Oded Maron and Tomás Lozano-Pérez, "A framework for multiple-instance learning," *Advances in neural information processing systems*, pp. 570–576, 1998. [Cited on pages 10, 19, 20, 21, 22, and 67.]

[30] Soumya Ray and Mark Craven, "Supervised versus multiple instance learning: An empirical comparison," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 697–704. [Cited on pages 10, 22, and 24.]

[31] Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li, "Multi-instance learning by treating instances as non-iid samples," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 1249–1256. [Cited on pages 10 and 28.]

[32] Jinbo Bi and Jianming Liang, "Multiple instance learning of pulmonary embolism detection with geodesic distance along vascular structure," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8. [Cited on page 10.]

[33] Alexander Vezhnevets and Joachim M Buhmann, "Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3249–3256. [Cited on page 10.]

[34] Changbo Yang, Ming Dong, and Farshad Fotouhi, "Region based image annotation through multiple-instance learning," in *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, 2005, pp. 435–438. [Cited on page 10.]

[35] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie, "Robust object tracking with online multiple instance learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1619–1632, 2011. [Cited on page 10.]

[36] Yuan Xie, Yanyun Qu, Cuihua Li, and Wensheng Zhang, "Online multiple instance gradient feature selection for robust visual tracking," *Pattern Recognition Letters*, vol. 33, no. 9, pp. 1075–1082, 2012. [Cited on page 10.]

[37] Zhe Lin, Gang Hua, and Larry S Davis, "Multiple instance feature for robust part-based object detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 405–412. [Cited on page 10.]

[38] Yixin Chen, Jinbo Bi, and James Z Wang, "Miles: Multiple-instance learning via embedded instance selection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 12, pp. 1931–1947, 2006. [Cited on pages 10, 23, 26, and 29.]

[39] Zhouyu Fu and Antonio Robles-Kelly, "An instance selection approach to multiple instance learning," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 911–918. [Cited on pages 10 and 22.]

[40] Sudheendra Vijayanarasimhan and Kristen Grauman, "Keywords to visual categories: Multiple-instance learning forweakly supervised object categorization," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8. [Cited on page 10.]

[41] Rouhollah Rahmani and Sally A Goldman, "Missl: Multiple-instance semi-supervised learning," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 705–712. [Cited on page 10.]

[42] Chengcui Zhang, Xin Chen, and Wei-Bang Chen, "An online multiple instance learning system for semantic image retrieval," in *Multimedia Workshops, 2007. ISMW'07. Ninth IEEE International Symposium on*. IEEE, 2007, pp. 83–84. [Cited on page 10.]

[43] Qi Zhang, Sally A Goldman, Wei Yu, and Jason E Fritts, "Content-based image retrieval using multiple-instance learning," in *ICML*. Citeseer, 2002, vol. 2, pp. 682–689. [Cited on pages 10, 28, and 68.]

[44] Zhouyu Fu, Antonio Robles-Kelly, and Jun Zhou, "Milis: Multiple instance learning with instance selection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 5, pp. 958–977, 2011. [Cited on pages 10, 29, 30, and 31.]

[45] David G Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004. [Cited on pages 11 and 62.]

[46] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 1, pp. 886–893. [Cited on pages 11, 34, 62, and 73.]

[47] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek, "Image classification with the fisher vector: Theory and practice," *International journal of computer vision*, vol. 105, no. 3, pp. 222–245, 2013. [Cited on pages 11, 73, and 74.]

[48] Li Deng, Dong Yu, et al., "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014. [Cited on page 11.]

[49] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105. [Cited on pages 11, 39, 43, and 79.]

[50] Geoffrey E Hinton, "Deep belief networks," *Scholarpedia*, vol. 4, no. 5, pp. 5947, 2009. [Cited on page 11.]

[51] Ruslan Salakhutdinov and Geoffrey E Hinton, "Deep boltzmann machines.," in *AISTATS*, 2009, vol. 1, p. 3. [Cited on page 11.]

[52] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015. [Cited on page 11.]

[53] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015. [Cited on pages 11, 46, and 86.]

[54] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724. [Cited on page 11.]

[55] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813. [Cited on page 11.]

[56] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*. Prague, 2004, vol. 1, pp. 1–2. [Cited on pages 16, 32, and 52.]

[57] Thomas Berg and Peter Belhumeur, "Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 955–962. [Cited on page 17.]

[58] Jia Deng, Jonathan Krause, and Li Fei-Fei, "Fine-grained crowdsourcing for fine-grained recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 580–587. [Cited on pages 17 and 53.]

[59] Catherine Wah, Grant Horn, Steve Branson, Subhransu Maji, Pietro Perona, and Serge Belongie, "Similarity comparisons for interactive fine-grained categorization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 859–866. [Cited on page 17.]

[60] Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie, "Visual recognition with humans in the loop," in *Computer Vision–ECCV 2010*, pp. 438–451. Springer, 2010. [Cited on page 17.]

[61] James Foulds and Eibe Frank, "A review of multi-instance learning assumptions," *The Knowledge Engineering Review*, vol. 25, no. 01, pp. 1–25, 2010. [Cited on page 18.]

[62] Qi Zhang and Sally A Goldman, "Em-dd: An improved multiple-instance learning technique," in *Advances in neural information processing systems*, 2001, pp. 1073–1080. [Cited on pages 19 and 22.]

[63] Wu-Jun Li and Dit-Yan Yeung, "Mild: Multiple-instance learning via disambiguation," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 1, pp. 76–89, 2010. [Cited on pages 20 and 29.]

[64] Cha Zhang, John C Platt, and Paul A Viola, "Multiple instance boosting for object detection," in *Advances in neural information processing systems*, 2005, pp. 1417–1424. [Cited on pages 21 and 83.]

[65] Cheng Yang and Tomas Lozano-Perez, "Image database retrieval with multiple-instance learning techniques," in *Data Engineering, 2000. Proceedings. 16th International Conference on*. IEEE, 2000, pp. 233–243. [Cited on pages 22, 28, and 68.]

[66] Andrew Karem and Hichem Frigui, "Fuzzy clustering of multiple instance data," in *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1–7. [Cited on page 22.]

[67] Yixin Chen and James Z Wang, "Image categorization by learning and reasoning with regions," *The Journal of Machine Learning Research*, vol. 5, pp. 913–939, 2004. [Cited on pages 23, 28, and 75.]

[68] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann, "Support vector machines for multiple-instance learning," in *Advances in neural information processing systems*, 2002, pp. 561–568. [Cited on pages 23, 24, 25, 75, and 91.]

[69] Thomas Gärtner, Peter A Flach, Adam Kowalczyk, and Alexander J Smola, "Multi-instance kernels.," in *ICML*, 2002, vol. 2, pp. 179–186. [Cited on page 23.]

[70] Mu Li, James T Kwok, and Bao-Liang Lu, "Online multiple instance learning with no regret," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1395–1401. [Cited on page 23.]

[71] Jun Wang and Jean-Daniel Zucker, "Solving multiple-instance problem: A lazy learning approach," 2000. [Cited on pages 23, 27, 75, and 91.]

[72] Qifan Wang, Luo Si, and Dan Zhang, "A discriminative data-dependent mixture-model approach for multiple instance learning in image classification," in *Computer Vision–ECCV 2012*, pp. 660–673. Springer, 2012. [Cited on page 23.]

[73] Nils Weidmann, Eibe Frank, and Bernhard Pfahringer, "A two-level learning method for generalized multi-instance problems," in *Machine Learning: ECML 2003*, pp. 468–479. Springer, 2003. [Cited on page 24.]

[74] Xin Xu and Eibe Frank, "Logistic regression and boosting for labeled bags of instances," in *Advances in knowledge discovery and data mining*, pp. 272–281. Springer, 2004. [Cited on page 24.]

[75] Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani, "1-norm support vector machines," *Advances in neural information processing systems*, vol. 16, no. 1, pp. 49–56, 2004. [Cited on pages 26 and 29.]

[76] James Foulds, *Learning instance weights in multi-instance learning*, Ph.D. thesis, Citeseer, 2008. [Cited on page 27.]

[77] Thomas Deselaers and Vittorio Ferrari, "A conditional random field for multiple-instance learning," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 287–294. [Cited on page 28.]

[78] Ruo Du, Qiang Wu, Xiangjian He, and Jie Yang, "Mil-skde: Multiple-instance learning with supervised kernel density estimation," *Signal Processing*, vol. 93, no. 6, pp. 1471–1484, 2013. [Cited on pages 29 and 31.]

[79] David J Crandall and Daniel P Huttenlocher, "Weakly supervised learning of part-based spatial models for visual object recognition," in *Computer Vision–ECCV 2006*, pp. 16–29. Springer, 2006. [Cited on page 31.]

[80] Ondřej Chum and Andrew Zisserman, "An exemplar model for learning object classes," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8. [Cited on page 31.]

[81] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari, "Weakly supervised localization and learning with generic knowledge," *International journal of computer vision*, vol. 100, no. 3, pp. 275–293, 2012. [Cited on page 31.]

[82] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid, "Multi-fold mil training for weakly supervised object localization," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 2409–2416. [Cited on page 32.]

[83] Shivani Agarwal and Dan Roth, "Learning a sparse representation for object detection," in *Computer Vision—ECCV 2002*, pp. 113–127. Springer, 2002. [Cited on page 32.]

[84] Cordelia Schmid et al., "Selection of scale-invariant parts for object class recognition," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 634–639. [Cited on page 32.]

[85] Zhi-Hua Zhou, Xiao-Bing Xue, and Yuan Jiang, "Locating regions of interest in cbir with multi-instance learning techniques," in *AI 2005: Advances in Artificial Intelligence*, pp. 92–101. Springer, 2005. [Cited on page 32.]

[86] Yu-Feng Li, James T Kwok, Ivor W Tsang, and Zhi-Hua Zhou, "A convex method for locating regions of interest with multi-instance learning," in *Machine learning and knowledge discovery in databases*, pp. 15–30. Springer, 2009. [Cited on page 32.]

[87] Mayank Juneja, Andrea Vedaldi, CV Jawahar, and Andrew Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 923–930. [Cited on page 32.]

[88] Efstratios Gavves, Basura Fernando, Cees GM Snoek, Arnold WM Smeulders, and Tinne Tuytelaars, "Local alignments for fine-grained categorization," *International Journal of Computer Vision*, vol. 111, no. 2, pp. 191–212, 2015. [Cited on page 32.]

[89] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*. ACM, 2004, vol. 23, pp. 309–314. [Cited on pages 32 and 69.]

[90] Carl Doersch, Abhinav Gupta, and Alexei A Efros, "Mid-level visual element discovery as discriminative mode seeking," in *Advances in neural information processing systems*, 2013, pp. 494–502. [Cited on page 32.]

[91] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik, "Detecting people using mutually consistent poselet activations," in *Computer Vision–ECCV 2010*, pp. 168–181. Springer, 2010. [Cited on page 33.]

[92] Ning Zhang, Ronan Farrell, Forrest Iandola, and Trevor Darrell, "Deformable part descriptors for fine-grained recognition and attribute prediction," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 729–736. [Cited on page 33.]

[93] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan, "Object detection with discriminatively trained part-based models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1627–1645, 2010. [Cited on pages 33 and 34.]

[94] Yuning Chai, Victor Lempitsky, and Andrew Zisserman, "Symbiotic segmentation and part localization for fine-grained categorization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 321–328. [Cited on page 33.]

[95] Christoph Göring, Erid Rodner, Alexander Freytag, and Joachim Denzler, "Nonparametric part transfer for fine-grained recognition," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 2489–2496. [Cited on page 33.]

[96] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell, "Part-based r-cnns for fine-grained category detection," in *Computer Vision–ECCV 2014*, pp. 834–849. Springer, 2014. [Cited on page 33.]

[97] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," *arXiv preprint arXiv:1310.1531*, 2013. [Cited on pages 33 and 73.]

[98] Marcel Simon and Erik Rodner, "Neural activation constellations: Unsupervised part model discovery with convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1143–1151. [Cited on page 33.]

[99] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al., "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2008–2016. [Cited on page 33.]

[100] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587. [Cited on page 33.]

[101] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99. [Cited on page 33.]

[102] Karel Lenc and Andrea Vedaldi, "R-cnn minus r," *arXiv preprint arXiv:1506.06981*, 2015. [Cited on page 33.]

[103] Robert Desimone and John Duncan, "Neural mechanisms of selective visual attention," *Annual review of neuroscience*, vol. 18, no. 1, pp. 193–222, 1995. [Cited on page 34.]

[104] Xiaoyu Wang, Ming Yang, Shenghuo Zhu, and Yuanqing Lin, "Regionlets for generic object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 17–24. [Cited on page 34.]

[105] Ali Borji and Laurent Itti, "State-of-the-art in visual attention modeling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 1, pp. 185–207, 2013. [Cited on page 34.]

[106] Derrick Parkhurst, Klinton Law, and Ernst Niebur, "Modeling the role of salience in the allocation of overt visual attention," *Vision research*, vol. 42, no. 1, pp. 107–123, 2002. [Cited on pages 34, 35, and 38.]

[107] Vidhya Navalpakkam and Laurent Itti, "An integrated model of top-down and bottom-up attention for optimizing detection speed," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. IEEE, 2006, vol. 2, pp. 2049–2056. [Cited on page 34.]

[108] Zhixiang Ren, Shenghua Gao, Liang-Tien Chia, and Ivor Wai-Hung Tsang, "Region-based saliency detection and its application in object recognition," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 24, no. 5, pp. 769–779, 2014. [Cited on page 34.]

[109] Junwei Han, King N Ngan, Mingjing Li, and Hong-Jiang Zhang, "Unsupervised extraction of visual attention objects in color images," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 16, no. 1, pp. 141–145, 2006. [Cited on page 34.]

[110] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik, "Contour detection and hierarchical image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 5, pp. 898–916, 2011. [Cited on page 34.]

[111] Yu-Fei Ma, Xian-Sheng Hua, Lie Lu, and Hong-Jiang Zhang, "User attention model based video summarization," *IEEE Transactions on Multimedia Journal*, 2004. [Cited on page 34.]

[112] Joydeep Ghosh, Yong Jae Lee, and Kristen Grauman, "Discovering important people and objects for egocentric video summarization," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1346–1353. [Cited on page 34.]

[113] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal, "Context-aware saliency detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 10, pp. 1915–1926, 2012. [Cited on pages 34 and 35.]

[114] Liang Li, Shuqiang Jiang, Zheng-Jun Zha, Zhipeng Wu, and Qingming Huang, "Partial-duplicate image retrieval via saliency-guided visual matching," *MultiMedia, IEEE*, vol. 20, no. 3, pp. 13–23, 2013. [Cited on page 34.]

[115] Songhe Feng, De Xu, and Xu Yang, "Attention-driven salient edge (s) and region (s) extraction with application to cbir," *Signal Processing*, vol. 90, no. 1, pp. 1–15, 2010. [Cited on page 34.]

[116] Wolfgang Einhäuser and Peter König, "Does luminance-contrast contribute to a saliency map for overt visual attention?," *European Journal of Neuroscience*, vol. 17, no. 5, pp. 1089–1097, 2003. [Cited on page 35.]

[117] Ali Borji, Dicky N Sihite, and Laurent Itti, "What stands out in a scene? a study of human explicit saliency judgment," *Vision research*, vol. 91, pp. 62–77, 2013. [Cited on page 35.]

[118] Kathryn Koehler, Fei Guo, Sheng Zhang, and Miguel P Eckstein, "What do saliency models predict?," *Journal of vision*, vol. 14, no. 3, pp. 14–14, 2014. [Cited on page 35.]

[119] Lior Elazary and Laurent Itti, "Interesting objects are visually salient," *Journal of vision*, vol. 8, no. 3, pp. 3–3, 2008. [Cited on page 35.]

[120] Christopher Michael Masciocchi, Stefan Mihalas, Derrick Parkhurst, and Ernst Niebur, "Everyone knows what is interesting: Salient locations which should be fixated," *Journal of vision*, vol. 9, no. 11, pp. 25–25, 2009. [Cited on page 35.]

[121] Roberto Valenti, Nicu Sebe, and Theo Gevers, "Image saliency by isocentric curvedness and color," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2185–2192. [Cited on page 35.]

[122] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum, "Learning to detect a salient object," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 2, pp. 353–367, 2011. [Cited on pages 35 and 37.]

[123] Ali Borji and Laurent Itti, "Exploiting local and global patch rarities for saliency detection," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 478–485. [Cited on page 35.]

[124] Gert Kootstra, Arco Nederveen, and Bart De Boer, "Paying attention to symmetry," in *British Machine Vision Conference (BMVC2008)*. The British Machine Vision Association and Society for Pattern Recognition, 2008, pp. 1115–1125. [Cited on page 35.]

[125] Xiaodi Hou, Jonathan Harel, and Christof Koch, "Image signature: Highlighting sparse salient regions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 1, pp. 194–201, 2012. [Cited on page 35.]

[126] Yin Li, Xiaodi Hou, Christof Koch, James Rehg, and Alan Yuille, "The secrets of salient object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 280–287. [Cited on page 35.]

[127] Xiaohui Shen and Ying Wu, "A unified approach to salient object detection via low rank matrix recovery," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 853–860. [Cited on page 35.]

[128] Ming Cheng, Niloy J Mitra, Xumin Huang, Philip HS Torr, and Song Hu, "Global contrast based salient region detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 3, pp. 569–582, 2015. [Cited on page 35.]

[129] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Tie Liu, Nanning Zheng, and Shipeng Li, "Automatic salient object segmentation based on context and shape prior.," in *BMVC*, 2011, vol. 6, p. 9. [Cited on page 36.]

[130] Christian Scharfenberger, Alexander Wong, Khalil Fergani, John Zelek, and David Clausi, "Statistical textural distinctiveness for salient region detection in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 979–986. [Cited on page 36.]

[131] Ming-Ming Cheng, Jonathan Warrell, Wen-Yan Lin, Shuai Zheng, Vibhav Vineet, and Nigel Crook, "Efficient salient region detection with soft image abstraction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1529–1536. [Cited on page 36.]

[132] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari, "What is an object?," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 73–80. [Cited on page 37.]

[133] Yangqing Jia and Mei Han, "Category-independent object-level saliency detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1761–1768. [Cited on page 37.]

[134] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun, "Saliency optimization from robust background detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2814–2821. [Cited on page 37.]

[135] Luca Marchesotti, Claudio Cifarelli, and Gabriela Csurka, "A framework for visual saliency detection with applications to image thumbnailing," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2232–2239. [Cited on page 38.]

[136] Kai-Yueh Chang, Tyng-Luh Liu, and Shang-Hong Lai, "From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 2129–2136. [Cited on page 38.]

[137] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li, "Salient object detection: A survey," *arXiv preprint arXiv:1411.5878*, 2014. [Cited on pages 38 and 73.]

[138] Wolf Kienzle, Matthias O Franz, Bernhard Schölkopf, and Felix A Wichmann, "Center-surround patterns emerge as optimal predictors for human saccade targets," *Journal of vision*, vol. 9, no. 5, pp. 7–7, 2009. [Cited on page 38.]

[139] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. [Cited on page 39.]

[140] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708. [Cited on page 39.]

[141] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*. IEEE, 2006, vol. 2, pp. 2169–2178. [Cited on pages 41 and 53.]

[142] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013. [Cited on page 43.]

[143] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9. [Cited on pages 43 and 44.]

[144] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. [Cited on pages 43, 45, 82, and 84.]

[145] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition.," in *Icml*, 2014, vol. 32, pp. 647–655. [Cited on page 45.]

[146] Sinno Jialin Pan and Qiang Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. [Cited on page 45.]

[147] Qiu-Lin Li and Jia-Feng He, "Vehicles detection based on three-frame-difference method and cross-entropy threshold method," *Computer Engineering*, vol. 37, no. 4, pp. 172–174, 2011. [Cited on page 46.]

[148] Wei Zhang, QM Jonathan Wu, Xiaokang Yang, and Xiangzhong Fang, "Multilevel framework to detect and handle vehicle occlusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 161–174, 2008. [Cited on page 46.]

[149] Ya Liu, Yao Lu, Qingxuan Shi, and Jianhua Ding, "Optical flow based urban road vehicle tracking," in *Computational Intelligence and Security (CIS), 2013 9th International Conference on*. IEEE, 2013, pp. 391–395. [Cited on page 46.]

[150] Alberto Faro, Daniela Giordano, and Concetto Spampinato, "Adaptive background modeling integrated with luminosity sensors and occlusion processing for reliable vehicle detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1398–1412, 2011. [Cited on page 46.]

[151] Yangqing Jia and Changshui Zhang, "Front-view vehicle detection by markov chain monte carlo method," *Pattern Recognition*, vol. 42, no. 3, pp. 313–321, 2009. [Cited on page 47.]

[152] Christos Tzomakas and Werner von Seelen, "Vehicle detection in traffic scenes using shadows," in *Ir-Ini, Institut fur Nueroinformatik, Ruhr-Universitat*. Citeseer, 1998. [Cited on page 47.]

[153] Soo Siang Teoh and Thomas Bräunl, "Symmetry-based monocular vehicle detection system," *Machine Vision and Applications*, vol. 23, no. 5, pp. 831–842, 2012. [Cited on page 47.]

[154] Junwen Wu, Xuegong Zhang, and Jie Zhou, "Vehicle detection in static road images with pca-and-wavelet-based classifier," in *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*. IEEE, 2001, pp. 740–744. [Cited on page 47.]

[155] Mustafa Ozuysal, Vincent Lepetit, and Pascal Fua, "Pose estimation for category specific multiview object localization," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 778–785. [Cited on page 47.]

[156] Hongliang Bai, Jianping Wu, and Changpin Liu, "Motion and haar-like features based vehicle detection," in *Multi-Media Modelling Conference Proceedings, 2006 12th International*. IEEE, 2006, pp. 4–pp. [Cited on page 47.]

[157] Bashirahamad F Momin and Tabssum M Mujawar, "Vehicle detection and attribute based search of vehicles in video surveillance system," in *Circuit, Power and Computing Technologies (ICCPCT), 2015 International Conference on*. IEEE, 2015, pp. 1–4. [Cited on page 47.]

[158] Michael Stark, Jonathan Krause, Bojan Pepik, David Meger, James J Little, Bernt Schiele, and Daphne Koller, "Fine-grained categorization for 3d scene understanding," *International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, 2011. [Cited on page 47.]

[159] Yan Li, Leon Gu, and Takeo Kanade, "Robustly aligning a shape model and its application to car alignment of unknown pose," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 9, pp. 1860–1876, 2011. [Cited on page 47.]

[160] Sayanan Sivaraman and Mohan Manubhai Trivedi, "Vehicle detection by independent parts for urban driver assistance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1597–1608, 2013. [Cited on page 47.]

[161] Norbert Buch, James Orwell, and Sergio A Velastin, "Detection and classification of vehicles for urban traffic scenes," 2008. [Cited on page 47.]

[162] Ying Shan, Harpreet S Sawhney, and Rakesh Kumar, "Unsupervised learning of discriminative edge measures for vehicle matching between nonoverlapping cameras," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 4, pp. 700–711, 2008. [Cited on page 48.]

[163] Zhen Dong, Yuwei Wu, Mingtao Pei, and Yunde Jia, "Vehicle type classification using a semisupervised convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2247–2256, 2015. [Cited on page 48.]

[164] Zhaoxiang Zhang, Tieniu Tan, Kaiqi Huang, and Yunhong Wang, "Three-dimensional deformable-model-based localization and recognition of road vehicles," *IEEE transactions on image processing*, vol. 21, no. 1, pp. 1–13, 2012. [Cited on page 48.]

[165] Jun-Wei Hsieh, Shih-Hao Yu, Yung-Sheng Chen, and Wen-Fong Hu, "Automatic traffic surveillance system for vehicle tracking and classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 2, pp. 175–187, 2006. [Cited on page 48.]

[166] Pablo Negri, Xavier Clady, Maurice Milgram, and Raphael Poulenard, "An oriented-contour point based voting algorithm for vehicle type classification," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*. IEEE, 2006, vol. 1, pp. 574–577. [Cited on pages 48, 50, and 52.]

[167] Ninad S Thakoor and Bir Bhanu, "Structural signatures for passenger vehicle classification in video," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1796–1805, 2013. [Cited on page 48.]

[168] Xiaoxu Ma and W Eric L Grimson, "Edge-based rich representation for vehicle classification," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. IEEE, 2005, vol. 2, pp. 1185–1192. [Cited on pages 48 and 86.]

[169] Mehran Kafai and Bir Bhanu, "Dynamic bayesian networks for vehicle classification in video," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 100–109, 2012. [Cited on page 48.]

[170] Fabrízia Medeiros de S Matos and Renata Maria Cardoso R de Souza, "Hierarchical classification of vehicle images using nn with conditional adaptive distance," in *International Conference on Neural Information Processing*. Springer, 2013, pp. 745–752. [Cited on page 48.]

[171] Matthew J Leotta and Joseph L Mundy, "Vehicle surveillance with a generic, adaptive, 3d vehicle model," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 7, pp. 1457–1469, 2011. [Cited on page 48.]

[172] Zezhi Chen, Tim Ellis, and Sergio A Velastin, "Vehicle type categorization: A comparison of classification schemes," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 74–79. [Cited on page 49.]

[173] Geoffrey McLachlan, *Discriminant analysis and statistical pattern recognition*, vol. 544, John Wiley & Sons, 2004. [Cited on page 49.]

[174] JM Bernardo and AFM Smith, "Bayesian theory wiley," *New York*, 1994. [Cited on page 49.]

[175] Vladimir Vapnik, *The nature of statistical learning theory*, Springer science & business media, 2013. [Cited on page 49.]

[176] Xavier Clady, Pablo Negri, Maurice Milgram, and Raphael Poulenard, "Multi-class vehicle type recognition system," in *Artificial Neural Networks in Pattern Recognition*, pp. 228–239. Springer, 2008. [Cited on pages 49, 50, 51, and 60.]

[177] Marian B Gorzalczany, *Computational intelligence systems and applications: neuro-fuzzy and fuzzy neural synergisms*, vol. 86, Physica, 2012. [Cited on page 49.]

[178] Hui-Zhen Gu and Suh-Yin Lee, "Car model recognition by utilizing symmetric property to overcome severe pose variation," *Machine vision and applications*, vol. 24, no. 2, pp. 255–274, 2013. [Cited on pages 50 and 51.]

[179] Margrit Betke, Esin Haritaoglu, and Larry S Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle," *Machine vision and applications*, vol. 12, no. 2, pp. 69–83, 2000. [Cited on page 50.]

[180] David Santos and Paulo Lobato Correia, "Car recognition based on back lights and rear view features," 2009. [Cited on page 50.]

[181] DF Llorca, D Colás, IG Daza, I Parra, and MA Sotelo, "Vehicle model recognition using geometry and appearance of car emblems from rear view images," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*. IEEE, 2014, pp. 3094–3099. [Cited on pages 50, 51, 55, and 60.]

[182] Greg Pearce and Nick Pears, "Automatic make and model recognition from frontal images of cars," in *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*. IEEE, 2011, pp. 373–378. [Cited on pages 50, 51, 60, and 86.]

[183] Daniel T Munroe and Michael G Madden, "Multi-class and single-class classification approaches to vehicle model recognition from images," *Proceedings of IEEE AICS*, 2005. [Cited on pages 50 and 60.]

[184] A Psyllos, Christos-Nikolaos Anagnostopoulos, and Eleftherios Kayafas, "Vehicle model recognition from frontal view image measurements," *Computer Standards & Interfaces*, vol. 33, no. 2, pp. 142–151, 2011. [Cited on pages 50, 60, and 62.]

[185] Li-Chih Chen, Jun-Wei Hsieh, Yilin Yan, and Bo-Yuan Wong, "Real-time vehicle make and model recognition from roads," in *2013 12th Conference on Information Technology and Applications in Outlying Islands*, 2013, pp. 1033–1040. [Cited on page 50.]

[186] Remigiusz Baran, Andrzej Glowacz, and Andrzej Matiolanski, "The efficient real-and non-real-time make and model recognition of cars," *Multimedia Tools and Applications*, vol. 74, no. 12, pp. 4269–4288, 2015. [Cited on pages 50 and 52.]

[187] Suhan Lee, Jeonghwan Gwak, and Moongu Jeon, "Vehicle model recognition in video," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 6, no. 2, pp. 175, 2013. [Cited on pages 50, 52, and 54.]

[188] Meena AbdelMaseeh, Islam Badreldin, Mohamed F Abdelkader, and Motaz El Saban, "Car make and model recognition combining global and local cues," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 910–913. [Cited on pages 50 and 60.]

[189] Vladimir S Petrovic and Timothy F Cootes, "Analysis of features for rigid structure vehicle type recognition.," in *BMVC*, 2004, pp. 1–10. [Cited on pages 50 and 60.]

[190] Farhad Mohamad Kazemi, Saeed Samadi, Hamid Reza Poorreza, and Mohamad-R Akbarzadeh-T, "Vehicle recognition based on fourier, wavelet and curvelet transforms-a comparative study," in *Information Technology, 2007. ITNG'07. Fourth International Conference on*. IEEE, 2007, pp. 939–940. [Cited on page 52.]

[191] Emmanuel J Candès and David L Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise c2 singularities," *Communications on pure and applied mathematics*, vol. 57, no. 2, pp. 219–266, 2004. [Cited on page 52.]

[192] Daniel Marcus Jang and Matthew Turk, "Car-rec: A real time car recognition system," in *applications of computer vision (WACV), 2011 IEEE Workshop on*. IEEE, 2011, pp. 599–605. [Cited on page 52.]

[193] Muhammad Fraz, Eran A Edirisinghe, and M Saquib Sarfraz, "Mid-level-representation based lexicon for vehicle make and model recognition," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 393–398. [Cited on pages 52 and 54.]

[194] Bailing Zhang, "Reliable classification of vehicle types based on cascade classifier ensembles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 322–332, 2013. [Cited on pages 52 and 53.]

[195] M Saquib Sarfraz, Ahmed Saeed, M Haris Khan, and Zahid Riaz, "Bayesian prior models for vehicle make and model recognition," in *Proceedings of the 7th International Conference on Frontiers of Information Technology*. ACM, 2009, p. 35. [Cited on page 53.]

[196] M Saquib Sarfraz and M Haris Khan, "A probabilistic framework for patch based vehicle type recognition.," in *VISAPP*, 2011, pp. 358–363. [Cited on page 53.]

[197] Mohammad Mahdi Arzani and Mansour Jamzad, "Car type recognition in highways based on wavelet and contourlet feature extraction," in *Signal and Image Processing (ICSIP), 2010 International Conference on*. IEEE, 2010, pp. 353–356. [Cited on page 53.]

[198] Jan Prokaj and Gérard Medioni, "3-d model based vehicle recognition," in *Applications of Computer Vision (WACV), 2009 Workshop on*. IEEE, 2009, pp. 1–7. [Cited on pages 53 and 60.]

[199] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei, "3d object representations for fine-grained categorization," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 554–561. [Cited on pages 53, 55, and 87.]

[200] Yen-Liang Lin, Vlad I Morariu, Winston Hsu, and Larry S Davis, "Jointly optimizing 3d model fitting and fine-grained classification," in *Computer Vision–ECCV 2014*, pp. 466–480. Springer, 2014. [Cited on pages 54, 55, 86, and 87.]

[201] Saining Xie, Tianbao Yang, Xiaoyu Wang, and Yuanqing Lin, "Hyper-class augmented and regularized deep learning for fine-grained image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2645–2654. [Cited on page 56.]

[202] Jakub Sochor, Adam Herout, and Jiri Havel, "Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3006–3015. [Cited on pages 56, 57, and 87.]

[203] Yongbin Gao and Hyo Jong Lee, "Deep learning of principal component for car model recognition," in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2015, p. 48. [Cited on page 56.]

[204] Yongbin Gao and Hyo Jong Lee, "Local tiled deep networks for recognition of vehicle make and model," *Sensors*, vol. 16, no. 2, pp. 226, 2016. [Cited on page 56.]

[205] Shaoyong Yu, Yun Wu, Wei Li, Zhijun Song, and Wenhua Zeng, "A model for fine-grained vehicle classification based on deep learning," *Neurocomputing*, 2017. [Cited on page 56.]

[206] Jie Fang, Yu Zhou, Yao Yu, and Sidan Du, "Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture," *IEEE Transactions on Intelligent Transportation Systems*, 2016. [Cited on pages 56 and 57.]

[207] Liang Liao, Ruimin Hu, Jun Xiao, Qi Wang, Jing Xiao, and Jun Chen, "Exploiting effects of parts in fine-grained categorization of vehicles," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 745–749. [Cited on page 58.]

[208] Hongsheng He, Zhenzhou Shao, and Jindong Tan, "Recognition of car makes and models from a single traffic-camera image," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3182–3192, 2015. [Cited on page 58.]

[209] Abdul Jabbar Siddiqui, Abdelhamid Mammeri, and Azzedine Boukerche, "Towards efficient vehicle classification in intelligent transportation systems," in *Proceedings of the 5th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*. ACM, 2015, pp. 19–25. [Cited on page 58.]

[210] Yu Zhang, Xiu-Shen Wei, Jianxin Wu, Jianfei Cai, Jiangbo Lu, Viet-Anh Nguyen, and Minh N Do, "Weakly supervised fine-grained categorization with part-based image representation," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1713–1725, 2016. [Cited on page 58.]

[211] Jonathan Krause, Timnit Gebru, Jia Deng, Li-Jia Li, and Li Fei-Fei, "Learning features and parts for fine-grained recognition," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 26–33. [Cited on page 59.]

[212] Kun Duan, Luca Marchesotti, and David J Crandall, "Attribute-based vehicle recognition using viewpoint-aware multiple instance svms," in *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*. IEEE, 2014, pp. 333–338. [Cited on page 59.]

[213] Vladimir Shapiro, Dimo Dimov, Stefan Bonchev, Veselin Velichkov, and Georgi Gluhchev, "Adaptive license plate image extraction," in *International Conference on Computer Systems and Technologies*, 2003, pp. 2–7. [Cited on page 60.]

[214] Sunghoon Kim, Daechul Kim, Younbok Ryu, and Gyeonghwan Kim, "A robust license-plate extraction method under complex image conditions," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. IEEE, 2002, vol. 3, pp. 216–219. [Cited on page 60.]

[215] Bai Hongliang and Liu Changping, "A hybrid license plate extraction method based on edge statistics and morphology," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. IEEE, 2004, vol. 2, pp. 831–834. [Cited on page 60.]

[216] Premnath Dubey, "Heuristic approach for license plate detection," in *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*. IEEE, 2005, pp. 366–370. [Cited on page 61.]

[217] Thanh-Tung Nguyen and Thuy Thi Nguyen, "A real time license plate detection system based on boosting learning algorithm," in *Image and Signal Processing (CISP), 2012 5th International Congress on*. IEEE, 2012, pp. 819–823. [Cited on page 61.]

[218] Christos Nikolaos E Anagnostopoulos, Ioannis E Anagnostopoulos, Vassilis Loumos, and Eleftherios Kayafas, "A license plate-recognition algorithm for intelligent transportation system applications," *IEEE Transactions on Intelligent transportation systems*, vol. 7, no. 3, pp. 377–392, 2006. [Cited on page 61.]

[219] Ching-Tang Hsieh, Yu-Shan Juan, and Kuo-Ming Hung, "Multiple license plate detection for complex background," in *Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on*. IEEE, 2005, vol. 2, pp. 389–392. [Cited on page 61.]

[220] Koen Van De Sande, Theo Gevers, and Cees Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010. [Cited on page 61.]

[221] Pan Chen, Xiang Bai, and Wenyu Liu, "Vehicle color recognition on urban road by feature context," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2340–2346, 2014. [Cited on page 61.]

[222] Nakhoon Baek, Sun-Mi Park, Ku-Jin Kim, and Seong-Bae Park, "Vehicle color classification based on the support vector machine method," in *International Conference on Intelligent Computing*. Springer, 2007, pp. 1133–1139. [Cited on page 61.]

[223] Jeong-Woo Son, Seong-Bae Park, and Ku-Jin Kim, "A convolution kernel method for color recognition," in *Advanced Language Processing and Web Information Technology, 2007. ALPIT 2007. Sixth International Conference on*. IEEE, 2007, pp. 242–247. [Cited on page 61.]

[224] Chuanping Hu, Xiang Bai, Li Qi, Pan Chen, Gengjian Xue, and Lin Mei, "Vehicle color recognition with spatial pyramid deep learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2925–2934, 2015. [Cited on page 62.]

[225] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen, "Face description with local binary patterns: Application to face recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 12, pp. 2037–2041, 2006. [Cited on page 62.]

[226] Chee Sun Won, Dong Kwon Park, and Soo-Jun Park, "Efficient use of mpeg-7 edge histogram descriptor," *Etri Journal*, vol. 24, no. 1, pp. 23–30, 2002. [Cited on page 62.]

[227] Wang Yunqiong, Liu Zhifang, and Xiao Fei, "A fast coarse-to-fine vehicle logo detection and recognition method," in *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*. IEEE, 2007, pp. 691–696. [Cited on page 62.]

[228] DF Llorca, R Arroyo, and MA Sotelo, "Vehicle logo recognition in traffic images using hog features and svm," in *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE, 2013, pp. 2229–2234. [Cited on pages 62, 63, and 76.]

[229] Apostolos Psyllos, Christos-Nikolaos Anagnostopoulos, and Eleftherios Kayafas, "M-sift: a new method for vehicle logo recognition," in *Vehicular Electronics and Safety (ICVES), 2012 IEEE International Conference on*. IEEE, 2012, pp. 261–266. [Cited on pages 62 and 77.]

[230] Apostolos P Psyllos, Christos-Nikolaos E Anagnostopoulos, and Eleftherios Kayafas, "Vehicle logo recognition using a sift-based enhanced matching scheme," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 2, pp. 322–328, 2010. [Cited on page 62.]

[231] Shuyuan Yu, Shibao Zheng, Hua Yang, and Longfei Liang, "Vehicle logo recognition based on bag-of-words," in *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*. IEEE, 2013, pp. 353–358. [Cited on page 62.]

[232] Yunqiong Wang, Zhifang Liu, and Fei Xiao, "A fast coarse-to-fine vehicle logo detection and recognition method," in *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*. IEEE, 2007, pp. 691–696. [Cited on page 62.]

[233] Jianli Xiao, Wenshu Xiang, and Yuncai Liu, "Vehicle logo recognition by weighted multi-class support vector machine ensembles based on sharpness histogram features," *IET Image Processing*, vol. 9, no. 7, pp. 527–534, 2015. [Cited on page 63.]

[234] Jim Kleban, Xing Xie, and Wei-Ying Ma, "Spatial pyramid mining for logo detection in natural scenes," in *Multimedia and Expo, 2008 IEEE International Conference on*. IEEE, 2008, pp. 1077–1080. [Cited on page 63.]

[235] Kai Zhou, Karthik Mahesh Varadarajan, Markus Vincze, and Fuqiang Liu, "Hybridization of appearance and symmetry for vehicle-logo localization," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. IEEE, 2012, pp. 1396–1401. [Cited on page 63.]

[236] Yue Huang, Ruiwen Wu, Ye Sun, Wei Wang, and Xinghao Ding, "Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1951–1960, 2015. [Cited on page 63.]

[237] Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, and Tiejun Huang, "Deep relative distance learning: Tell the difference between similar vehicles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2167–2175. [Cited on pages 63 and 87.]

[238] Rogerio Schmidt Feris, Behjat Siddiquie, James Petterson, Yun Zhai, Ankur Datta, Lisa M Brown, and Sharath Pankanti, "Large-scale vehicle detection, indexing, and search in urban surveillance videos," *IEEE Transactions on Multimedia*, vol. 14, no. 1, pp. 28–42, 2012. [Cited on page 63.]

[239] Dominik Zapletal and Adam Herout, "Vehicle re-identification for automatic video traffic surveillance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 25–31. [Cited on page 63.]

[240] Xinchen Liu, Wu Liu, Huadong Ma, and Huiyuan Fu, "Large-scale vehicle re-identification in urban surveillance videos," in *Multimedia and Expo (ICME), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6. [Cited on page 64.]

[241] Katerina Fragkiadaki and Jianbo Shi, "Figure-ground image segmentation helps weakly-supervised learning of objects," in *Computer Vision–ECCV 2010*, pp. 561–574. Springer, 2010. [Cited on page 68.]

[242] Carolina Galleguillos, Boris Babenko, Andrew Rabinovich, and Serge Belongie, *Computer Vision – ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part I*, chapter Weakly Supervised Object Localization with Stable Segmentations, pp. 193–207, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. [Cited on page 68.]

[243] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu, "An efficient k-means clustering algorithm: Analysis

and implementation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 881–892, 2002. [Cited on page 69.]

[244] Dorin Comaniciu and Peter Meer, "Mean shift: A robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002. [Cited on page 69.]

[245] Anil K Jain and Farshid Farrokhnia, "Unsupervised texture segmentation using gabor filters," in *Systems, Man and Cybernetics, 1990. Conference Proceedings., IEEE International Conference on*. IEEE, 1990, pp. 14–19. [Cited on page 69.]

[246] Anne M Treisman and Garry Gelade, "A feature-integration theory of attention," *Cognitive psychology*, vol. 12, no. 1, pp. 97–136, 1980. [Cited on page 70.]

[247] Jeremy M Wolfe and Todd S Horowitz, "What attributes guide the deployment of visual attention and how do they do it?," *Nature reviews neuroscience*, vol. 5, no. 6, pp. 495–501, 2004. [Cited on pages 70 and 71.]

[248] Stephen E Palmer, *Vision science: Photons to phenomenology*, vol. 1, MIT press Cambridge, MA, 1999. [Cited on pages 70 and 71.]

[249] Edgar Rubin et al., "Figure and ground," *Readings in perception*, pp. 194–203, 1958. [Cited on page 71.]

[250] Jianming Zhang and Stan Sclaroff, "Exploiting surroundedness for saliency detection: a boolean map approach," 2015. [Cited on page 71.]

[251] Liqiang Huang and Harold Pashler, "A boolean map theory of visual attention.," *Psychological review*, vol. 114, no. 3, pp. 599, 2007. [Cited on page 71.]

[252] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International journal of computer vision*, vol. 73, no. 2, pp. 213–238, 2007. [Cited on page 73.]

[253] Anelia Angelova and Shenghuo Zhu, "Efficient object detection and segmentation for fine-grained recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 811–818. [Cited on page 73.]

[254] Jan Theeuwes, "Visual selective attention: A theoretical analysis," *Acta psychologica*, vol. 83, no. 2, pp. 93–154, 1993. [Cited on page 76.]

[255] Michelangelo Diligenti, Marco Gori, Marco Maggini, and Enrico Martinelli, "Adaptive graphical pattern recognition for the classification of company logos," *Pattern Recognition*, vol. 34, no. 10, pp. 2049–2061, 2001. [Cited on page 76.]

[256] Dana H Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern recognition*, vol. 13, no. 2, pp. 111–122, 1981. [Cited on page 77.]

[257] Martin A Fischler and Robert C Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981. [Cited on page 78.]

[258] Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann L Cun, "Learning convolutional feature hierarchies for visual recognition," in *Advances in neural information processing systems*, 2010, pp. 1090–1098. [Cited on page 80.]

[259] L Dlagnekov and S Belongie, "Ucsd/calit2 car license plate make and model database," 2005. [Cited on page 86.]

[260] Iffat Zafar, Eran A Edirisinghe, S Acar, and Helmut E Bez, "Two-dimensional statistical linear discriminant analysis for real-time robust vehicle-type recognition," in *Electronic Imaging 2007*. International Society for Optics and Photonics, 2007, pp. 649602–649602. [Cited on page 86.]

[261] Zhen Dong, Yuwei Wu, Mingtao Pei, and Yunde Jia, "Vehicle type classification using a semisupervised convolutional neural network," p. in press, 2015. [Cited on page 86.]

[262] Constantine P Papageorgiou and Tomaso Poggio, "A trainable object detection system: Car detection in static images," 1999. [Cited on page 86.]

[263] Bastian Leibe, Nico Cornelis, Kurt Cornelis, and Luc Van Gool, "Dynamic 3d scene analysis from a moving vehicle," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8. [Cited on page 86.]

[264] Kevin Matzen and Noah Snavely, "Nyc3dcars: A dataset of 3d vehicles in geographic context," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 761–768. [Cited on page 86.]

[265] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, 2011, number EPFL-CONF-192376. [Cited on page 99.]

[266] Ian J Goodfellow, David Warde-Farley, Pascal Lamblin, Vincent Dumoulin, Mehdi Mirza, Razvan Pascanu, James Bergstra, Frédéric Bastien, and Yoshua Bengio, "Pylearn2: a machine learning research library," *arXiv preprint arXiv:1308.4214*, 2013. [Cited on page 99.]

[267] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678. [Cited on page 99.]

[268] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008. [Cited on page 101.]

[269] Joseph Redmon and Ali Farhadi, "Yolo9000: Better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016. [Cited on page 106.]

[270] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*. Springer, 2014, pp. 740–755. [Cited on page 106.]

CURRICULUM VITAE

NAME: Faezeh Tafazzoli

ADDRESS: Computer Engineering and Computer Science Department
Speed School of Engineering, University of Louisville
Louisville, KY 40292

EDUCATION:

Ph.D., Computer Science & Engineering, April 2017
University of Louisville, *Louisville, Kentucky*

M.Sc., Computer Science & Engineering, May 2012
University of Nevada, Reno, *Nevada, Reno*

M.Sc., Computer Engineering, May 2008
Amirakabir University of Technology, *Tehran, Iran*

CONFERENCE PUBLICATIONS:

1. **F. Tafazzoli** and H. Frigui, *"A Multiple Instance Learning Approach to Vehicle Make and Model Recognition"*, 8th International Conference of Pattern Recognition Systems (ICPRS), Madrid, Spain, July 2017.

2. **F. Tafazzoli** and H. Frigui, *"Vehicle Make and Model Recognition for Automated Vehicular Surveillance"*, Women in Computer Vision (WiCV), CVPR, Honolulu, HI, July 2017.

3. **F. Tafazzoli** and H. Frigui, *"Vehicle Make and Model Recognition Using Local Features and Logo Detection"*, 8th International IEEE Symposium on Signal, Image,

Video and Comminucations (ISIVC), Tunis, Tunisia, November 2016.

4. **F. Tafazzoli** and H. Frigui, *"Saliency-based Multiple Instance Framework for Vehicle Make and Model Recognition"*, Women in Computer Vision (WiCV), CVPR, Las Vegas, NV, June 2016.

5. **F. Tafazzoli**, M. N. Saadatzi, K. C. Welch and J. Graham, *"EmotiGO: Bluetooth-enabled Eyewear for Unobtrusive Physiology-based Emotion Recognition"*, IEEE Conference on Automation Science and Engineering (CASE), Fort Worth, Texas, August 2016.

6. **F. Tafazzoli**, G. Bebis, S. Louis and M. Hussain, *"Improving Human Gait Recognition Using Feature Selection"*, Advances in Visual Computing (ISVC), Las Vegas, NV, December 2014.

JOURNAL PUBLICATIONS:

1. **F. Tafazzoli**, G. Bebis, S. Louis and M. Hussain, *"Genetic Feature Selection for Gait Recognition"*, Journal of Electronic Imaging, February 2015.

2. **F. Tafazzoli** and R. Safabakhsh, *"Model-Based Human Gait Recognition Using Leg and Arm Movements"*, Engineering Applications of Artificial Intelligence, December 2010.

PATENTS:

1. **F. Tafazzoli**, B. Xu, H. Wu and R. Loce, *"Automatic Visual Remote Assessment of Movement Symptoms in People with Parkinson's Disease for MDS-UPDRS Finger Tapping Task"*, US Patent # 20160089073, March 2016.

2. **F. Tafazzoli**, B. Xu, W. Wu and R. Loce, *"Automatic Frontal-View Gait Segmentation for Abnormal Gait Quantification"*, Pending US Patent.

RESEARCH AND WORK EXPERIENCE:

1. **PARC**, Rochester, NY, *May 2015-December 2015*

   Research scientist Intern, Video and Image Analytics Lab.

2. **Xerox Innovation Group (XRCW)**, Webster, NY, *May 2014-August 2014*

   Research scientist Intern, Systems Lab.

3. **Eye-Com Corporation**, Reno, NV, *May 2011-September 2011*

   Software developer Intern.

HONORS AND AWARDS:

1. Graduate Dean's Citation Award, UofL, April 2017
2. CECS Arthur M. Riehl Award, UofL, April 2017
3. NSF Innovation-Corps Award, UofL, August 2016
4. Women in Computer Vision Scholarship, CVPR, 2016
5. CSE Doctoral Award, UofL, 2016
6. Martha & Frank Diebold Award, UofL, 2016
7. 1st place Graduate Research Award, 101st Kentucky Academy of Science, 2015
8. Speed Up Entrepreneurial Program Award, UofL, 2015
9. Best Poster, UofL Graduate Research Symposium, 2014
10. Best Graduate Poster Award, ACM-W, Ky-triwic, 2014
11. Grace Hopper Scholarship, 2014, 2015
12. CRA-W Graduate Cohort Scholarship, 2013